

UNIVERZITA KARLOVA V PRAZE

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie

Studijní program: Geografie (navazující magisterské studium)

Studijní obor: Kartografie a geoinformatika



Bc. Jakub Hettler

**AUTOMATICKÁ PUBLIKACE METADAT A DAT PRO
MAPOVÉ A KATALOGOVÉ
SYSTÉMY Z RASTROVÝCH PODKLADŮ V
POSTGRESQL**

**AUTOMATIC PUBLICATION OF DATA AND METADATA FOR
MAP AND CATALOGUE SYSTEMS FROM RASTER SOURCES
IN POSTGRESQL**

Diplomová práce

Vedoucí diplomové práce: Mgr. Lukáš Brůha

Praha 2012

Zadání diplomové práce

pro Bc. Jakuba Hettlera

obor Kartografie a geoinformatika

Název tématu: Automatická publikace metadat a dat pro mapové a katalogové systémy z rastrových podkladů v PostgreSQL

Zásady pro vypracování

Rostoucí objem dat spravovaných v rámci systémů infrastruktury prostorových dat (SDI), především kontinuální tvorba rastrů jako jsou družicové snímky, klade požadavky na automatizaci procesů správy a efektivnější využívání prostředků jednotlivých částí SDI.

Cílem práce je připravit postupy a nástroje pro zjišťování informací o a ze souboru rastrových dat uložených v databázi prostřednictvím mapových a katalogových služeb, a to i v kontextu ostatních relevantních dat, která jsou v SDI k dispozici. Praktická implementace bude vytvořena nad databázovou platformou PostgreSQL a aplikací GeoNetwork opensource. Pro tyto účely:

- bude navržen a prakticky ověřen způsob uložení rastru v PostgreSQL s využitím rastrové podpory Postgis (PgRaster) a databázový nástroj (procedura, trigger), který bude zajišťovat přípravu a aktuálnost metadat pro katalogový server,- bude navržen a odzkoušen postup automatické extrakce metadat uložených v PostgreSQL do systému GeoNetwork opensource,
- dalším aspektem řešení bude návrh metodiky sestavování mapových kompozic na základě uživatelských dotazů,
- časové odezvy mapových služeb, dodržení standardů OGC a vhodnost mapových kompozic implementovaného systému budou testovány na datech z MERIS v kontextu dalších datových sad v systému fakultní SDI. Testování bude provedeno pomocí definovaných uživatelských dotazů s prostorovými kritérii a na možnostech využití v mapové kompozici

Rozsah grafických prací: dle potřeby

Rozsah průvodní zprávy: 60-80 stran

Seznam odborné literatury:

RACINE, Pierre. *Store, manipulate and analyze raster data within the PostgreSQL/PostGIS spatial database*. In Sborník konference FOSS4G Denver [online]., 2011 [cit. 2011-10-30]. Dostupné z URL: <http://trac.osgeo.org/postgis/raw-attachment/wiki/WKTRaster/PierreRacine_FOSS4G-2011.pdf>.

OBE, Regina; HSU, Leo. *PostGIS in Action*. Stamford : Manning Publications Co., 2011. 520 s. ISBN 9781935182269.

Giannecchini, Simone; Romagnoli, Daniele. *Raster data in GeoServer and GeoTools: Achievements, issues and future developments*. In Sborník konference FOSS4G Denver [online]., 2011 [cit. 2011-10-30]. Dostupné z URL: <http://demo.geo-solutions.it/share/foss4g2011/gt_raster_simone_foss4g2011.pdf>.

GRILL, Stanislav; JEDLIČKA, Jan; SCHNEIDER, Michal; ŠTEFANOVÁ, Eva. *Archive and catalogue system for receiving satellite data as a part of academic SDI* [online]. Praha : IOS Press Books Online, 2010. 10 s. PřF UK Praha, Katedra aplikované geoinformatiky a kartografie. Dostupné z URL: <<http://www.booksonline.iospress.nl/Content/View.aspx?piid=16427>>. ISBN 978-1-60750-493-1.

Vedoucí diplomové práce: Mgr. Lukáš Brůha

Konzultant diplomové práce:

Datum zadání diplomové práce: 29. 11. 2011

Termín odevzdání diplomové práce: jaro 2012

Platnost tohoto zadání je po dobu jednoho akademického roku.

.....
Vedoucí diplomové práce

.....
Vedoucí katedry

V Praze dne 29. 11. 2011

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a že jsem všechny použité prameny řádně citoval.

Jsem si vědom toho, že případné použití výsledků, získaných v této práci, mimo Univerzitu Karlovu v Praze je možné pouze po písemném souhlasu této univerzity.

Svoluji k zapůjčení této práce pro studijní účely a souhlasím s tím, aby byla řádně vedena v evidenci vypůjčovatelů.

V Praze dne 29.8.2012

.....
Jakub Hettler

Poděkování

Na tomto místě bych chtěl poděkovat Mgr. Lukáši Brůhovi za vynikající spolupráci, čas věnovaný konzultacím a podnětným připomínkám a hlavně přípravě samotného tématu. Velký dík patří mému stávajícímu zaměstnavateli, který mi umožnil proniknout do komerční sféry GIS a získat mnohé zkušenosti, které jsou na akademické půdě velkým přínosem. Nakonec bych chtěl poděkovat mé rodině, za podporu během celého studia a přátelům, kteří byli mnohdy zdrojem zajímavých myšlenek a kritických názorů, které často vedly ke zdárnému dosažení cíle.

Automatická publikace metadat a dat pro katalogové a mapové systémy z PostgreSQL

Abstrakt

Cílem práce je návrh a realizace funkčního řešení aplikace pro automatickou publikaci rastrových dat a metadat z databáze PostgreSQL do mapových a katalogových služeb s použitím open source technologií.

Základem aplikace je PostgreSQL s prostorovými nadstavbami PostGIS a PostGIS raster, které slouží pro uložení rastrových dat i metadat. Práce hodnotí možnosti uložení rastrů v databázi z různých pohledů např. vhodnost uložení pro analýzy vs. uložení pro publikaci. Je navržena struktura pro uložení rastrů umožňující co nejširší využití rastrů na straně mapových služeb i pro analytické účely nad importovanými rastry. Další částí je průzkum možností importních nástrojů rastrů do databáze a navržení vlastního importu.

Jsou zhodnoceny možnosti open source technologií a standardů pro vytvoření celého řešení. Jako metadatové řešení je využito GeoNetwork opensource, jako mapový server GeoServer.

Práce hodnotí výslednou aplikaci, popisuje přínosy a nedostatky, u kterých případně navrhuje řešení. Aplikace je testována nad reálnými daty v prostředí GeoNetwork opensource, kde jsou dostupná metadata i výstupy mapových služeb.

Klíčová slova: PostGIS, PostGIS raster, GeoServer, GeoNetwork opensource, metadata, webové mapové služby, OGC, Java, automatická publikace

Automatic publication of data and metadata for map and catalogue systems from raster sources in PostgreSQL

Abstract

The main goal of the presented work is the design and implementation of the application for the automatic publication of the raster data and metadata from the PostgreSQL database to the map and catalog services. The application should exclusively utilize the open source software and technologies.

The fundamental component of the developed application is the PostgreSQL database with the PostGIS and PostGIS raster extensions. The presented work evaluates the possibilities of the raster storage from different points of view -e.g. the suitability for the further data processing or the publication of the raster data. The most suitable structure for raster storage is then proposed with respect to analytical and publication usage of the stored data.

The possibilities of the open source software for the solution and implementation of the presented problem are then inspected. The GeoNetwork and GeoServer are utilized as a metadata and map server solution. The results of the deployment of these technologies is then evaluated for the real world data and compared with other available related solutions.

Keywords: PostGIS, PostGIS raster, GeoServer, GeoNetwork opensource, metadata, web map services, OGC, Java, automatic publication

OBSAH

Kapitola 1	
Úvod	12
1.1. Cíle práce	13
Kapitola 2	
Dostupné technologie, standardy a jejich zhodnocení	15
2.1. OGC a ISO	16
2.1.1. OWS	17
2.1.1.1. WMS	17
2.1.1.2. WFS	18
2.1.1.3. WMTS	18
2.1.1.4. CSW a ISO standardy metadat	19
2.1.2. WMC	20
2.1.3. SFS	20
2.1.4. KML	20
2.1.5. GML	20
2.2. Open source a svobodný software	21
2.2.1. Nejrozšířenější open source licence	22
2.3. Open source GIS	23
2.3.1. Open Source Geospatial Foundation	23
2.3.2. FOSS4G software list	24
2.3.3. Ostatní zdroje open source a svobodného GIS	24
2.4. Proprietární GIS	25
2.5. Komponenty SDI pro automatickou publikaci dat a metadat z PostgreSQL	25
2.5.1. Prostorové databázové systémy s podporou rastrových dat (SDBMS)	26
2.5.1.1. PostgreSQL/PostGIS/PostGIS raster	26
2.5.1.2. SQLite/SpatialLite/RasterLite	27
2.5.1.3. Oracle/Oracle Spatial	27
2.5.1.4. Ostatní SDBMS (bez podpory rastrových typů)	28
2.5.2. GIS knihovny a rozšíření	28
2.5.2.1. GDAL/OGR (Geospatial Data Abstraction Library)	28
2.5.2.2. Proj.4	29
2.5.2.3. GEOS (Geometry Engine, Open Source)	29
2.5.2.4. JTS (Java topology suite)	29
2.5.2.5. GeoTools	30
2.5.3. Desktopové GIS	31
2.5.4. Mapové servery	32
2.5.5. Weboví GIS klienti	34
2.5.6. Metadatová řešení	34

Kapitola 3	
Koncept a struktura řešení	36
3.1. Možnosti uložení rastrů v PostgreSQL	37
3.1.1. in-database	37
3.1.2. out-of-database	38
3.1.3. Obecné zhodnocení na úrovni uložení rastrů v PostgreSQL	39
3.2. Návrh řešení a definice dílčích problémů	39
Kapitola 4	
Implementační prostředí	43
4.1. Databázové prostředí (PostgreSQL s prostorovou nadstavbou PostGIS)	44
4.1.1. Instalace PostgreSQL 9.1 +	44
4.1.2. Instalace PostGIS 2.0.0 s podporou PostGIS raster	44
4.1.2.1. Konfigurace databáze s podporou PostGIS	45
4.2. Mapový server (GeoServer 2.1.4 +)	47
4.2.1. Instalace GeoServeru	47
4.2.2. Instalace podpory PostGIS raster do GeoServeru	47
4.3. Katalogový server/klient GeoNetwork	48
4.3.1. Instalace GeoNetworku s podporou PostgreSQL	48
4.4. Desktopový klient a závislosti	49
Kapitola 5	
Vývoj programu MTDTRasPub	51
5.1. Ověření manuální cestou	51
5.2. Popis funkcionality a testování	52
5.2.1. Vstupní data a podmínky	52
5.2.2. Metadata	52
5.2.3. Datové struktury pro rastry	53
5.2.4. Publikace mapové služby	54
5.2.5. Uživatelské kompozice	55
5.2.6. Výstup	55
5.3. Testování	56
Kapitola 6	
Diskuze	59
Kapitola 7	
Závěr	62
Seznam zdrojů informací	63
Seznam příloh	68

PŘEHLED POUŽITÝCH ZKRATEK

API	A pplication P rogramming I nterface
BLOB	B inary L arge O bject
EPSG	E uropean P etroleum S urvey G roup
FOSS	F ree and O pen S ource S oftware
FOSS4G	F ree and O pen S ource S oftware for G eospatial
FSF	F ree S oftware F oundation
GEOS	G eospatial E ngine, O pen S ource
GDAL	G eospatial D ata A bstraction L ibrary
GIS	G eografický i nformační s ystém
GiST	G eneralized S earch T ree
GML	G eography M arkup L anguage
ČÚZK	Č eský ú řad z eměměřický a k atastrální
HTTP	H yper T ext T ransfer P rotocol
JAI	J ava A dvanced I maging
JRE	J ava R untime E nvironment
JTS	J ava T opology S uite
IMJDBC	I mage M osaic J ava D atabase C onnectivity
ISO	I nternational O rganization for S tandardization
KAGIK	K atedra a plikované g eoinformatiky a k artografie
KML	K eyhole M arkup L anguage
MBR	M inimum B ounding R ectangle
NTS	N et T opology S uite
OGC	O pen G eospatial C onsorcium
OSGeo	O pen S ource G eospatial F oundation

OSI	O pen S ource I nitiative
OSS	O pen S ource S oftware
OWS	O pen G IS W eb S ervices C ommon S pecification
PNG	P ortable N etwork G raphic
REST	R Epresentational S tate T ransfer
SDBMS	S patial D atabase M anagement S ystem
SDI	S patial D ata I nfrastructure
SFS	S imple F eature S pecification
UUID	U niversally U nique I dentifier
W3C	W orld W ide W eb C onsorcium
WCS	W eb C overage S ervice
WFS	W eb F eature S ervice
WMS	W eb M ap S ervice
WMTS	W eb M ap T ile S ervice
XML	e X tensible M arkup L anguage
ZABAGED	Z ákladní b áze g eografických d at

SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1: Přehled typů GIS	16
Obrázek 2: Schéma požadavku GetMap WMS.....	17
Obrázek 3: Vztah kategorie software podle licencí	22
Obrázek 4: Výběr z FOSS4G projektů v roce 2011	24
Obrázek 5: Obecný koncept fungování aplikace MTDTRasPub	25
Obrázek 6: Ukázky prostorových predikátů.....	30
Obrázek 7: Počet uživatelů přispívajících do mailing listu (2008 - 2010).....	32
Obrázek 8: Počet vývojářů přispívajících do dev-mailing listu (2008 - 2010)	32
Obrázek 9: Výsledky hodnocení propustnosti mapových serverů na počtu klientů	33
Obrázek 10: Příklad pyramid nad rastrem	37
Obrázek 11: Návrh aplikace MTDTRasPub	40
Obrázek 12: Přesahy rastru vytvářející rám okolo publikovaného rastru	58
Tabulka 1: Požadavky na databázi PostgreSQL	26
Tabulka 2: Požadavky na databázi SQLite	27
Tabulka 3: Požadavky na databázi Oracle	28
Tabulka 4: Použité GIS knihovny	30
Tabulka 5: Výsledek dotazu pro ověření instalace PostGIS	45
Tabulka 6: Struktura tabulek v PostGIS po publikaci IMJDBC – mateřská tabulka	54
Tabulka 7: Struktura tabulek v PostGIS - tabulka o_2_marklissa	54
Tabulka 8: Testovaná data	56
Tabulka 9: Výsledky testování časových odezev v sekundách.....	56

KAPITOLA 1

Úvod

Stále zrychlující tempo dnešní společnosti na sebe dlouho nenechává čekat ani technologie, které produkují stále větší množství dat, ať již ve formě popisných informací (atributů) nebo pro obory geoinformačních systémů (dále jen GIS) specifická prostorová data. Pryč jsou časy, kdy rozsáhlé datové sklady byly pouze dominantou autorů či vlastníků dat, případně různých specializovaných archivů. Stále častěji mají vlastní sbírky prostorových dat značných rozměrů instituce v podobě škol, vědeckých ústavů, správních orgánů nebo v neposlední řadě například studenti GIS specializací, kteří po dobu svého studia nastřádají nepřeborné množství dat různého typu.

S vývojem technologií pro sběr a zpracování dat, s tím související významné zvětšování jejich objemu a stále větší dostupnosti běžnému uživateli, ovšem vzniká mnoho vedlejších úkolů, které je potřebné v návaznosti na tyto skutečnosti řešit.

Pomineme-li samotný sběr a pořizování, nastává hned vzápětí další série otázek, jak data „přivést k životu“, aby nezůstala v archivech pořizující instituce a dostala se ke koncovému uživateli, pokud se jedná o data, která se mají ze své podstaty dostat i mimo hranice pořizující organizace.

Jednou z prvních otázek je, jak data ukládat v databázových systémech. Jako další úkol je třeba zodpovědět otázku, jaké jsou možnosti publikace samotných dat do mapových služeb. Pokud ale vypublikujeme samotná data, budou velmi těžko dohledatelná a identifikovatelná v množství ostatních informací, která jsou dnes dostupná, proto je nutné si položit ještě následující otázku. Jak vhodně publikovat metadata o datech, která byla vypublikována jako mapová služba. Nakonec, vzhledem k velkým objemům a případným kontinuálním proudům dat (např. v případě radarových snímků, velkého množství historických map, ...), je třeba se zamyslet nad automatizací celého procesu, která je podstatnou součástí řešení.

Protože je cílem, aby tento přístup byl multiplatformní, co nejlevnější, splňující zadané požadavky a snadno začlenitelný jako součást SDI Katedry aplikované geoinformatiky

a kartografie UK v Praze (dále jen KAGIK), jsou používány výhradně open source a volně dostupné technologie.

Tato práce se bude zabývat pouze rastrovými daty a to vzhledem k následujícím faktům. O vektorových datech bylo napsáno mnoho publikací a jsou léta užívána a propojení databázových systémů a mapových serverů je vyřešenou otázkou. U rastrových dat ve spojení s open source databází a mapovým serverem by se dalo hovořit stále jako o oblasti s řadou otevřených problémů, jak tato práce v některých ohledech ukazuje a jak také ukazuje vývoj samotné nadstavby PostGIS raster v posledních letech, kdy první verze vyšla v březnu 2009 (Arévalo, 2010).

Práci můžeme rozdělit na dva velké tematické celky. Na část teoretickou, která se zabývá možnostmi, popisem a momentálními výsledky prací v odvětví publikace rastrových dat a metadat z PostgreSQL a reflektuje aktuální stav, ze kterého by měla vycházet část praktická.

Praktická část popisuje samotné implementační prostředí a konkrétní zvolené technologie tak, aby byl čtenář schopen proniknout do logiky aplikace a bez problémů ji tak uchopit a případně použít ve vlastní implementaci. Zároveň popisuje aplikaci samotnou a do detailu rozebírá jednotlivé kroky a možnosti aplikace.

1.1. Cíle práce

Cílem práce je navrhnout a realizovat funkční řešení pro automatickou publikaci rastrových dat a jejich metadat z databázového systému PostgreSQL do mapových a katalogových služeb.

Aby bylo možné postavit funkční základ celé aplikace, bude nutné prostudovat možnosti PostgreSQL s jeho prostorovými nadstavbami PostGIS a PostGIS raster. Ty budou základním stavebním kamenem pro uložení rastrových dat a případných dalších vektorových dat, které se budou ve finální struktuře SDI jistě vyskytovat, a také příslušných metadat. Budou zhodnoceny možnosti uložení rastrů v databázi z různých pohledů např. vhodnost uložení pro analýzy vs. vhodnost uložení pro publikaci. Práce navrhne strukturu pro uložení rastrů, aby bylo možné co nejširší využití rastrů jak na straně využití s mapovými službami, tak pro analytické účely nad importovanými rastry. Budou také zkoumány možnosti importních nástrojů rastrů a případně navržen vlastní import.

Protože je cílem, aby aplikace byla co nejvíce „otevřená“, bude se práce zabývat možnostmi open source mapových serverů, které mají schopnost komunikace s datovým typem PostGIS raster a bude porovnána také jejich vhodnost využití vzhledem k využití v různých typech GIS aplikací. V této kapitole bude zároveň probrána problematika publikace uživatelských sestavení (možnosti tvorby skupin vrstev s možností uložení). Budou řešeny možnosti publikace rastrových dat vzhledem ke zvolené technologii, popsána omezení a navržena jejich řešení. Bude také nutné tento proces automatizovat. Tento úkol nebyl ještě v literatuře řešen.

Abychom byli schopni publikace metadat a mohli využít všechny dostupné technologie, bude uveden přehled open source možností na poli metadatových klientů a katalogových serverů. Zde

bude použito metadatové řešení GeoNetwork opensource. Pro automatizaci tohoto procesu bude využita některá z podporovaných možností pro ovládání GeoNetwork opensource, například ve formě GeoNetwork XML services, kde bude přihlíženo k OGC standardům.

Budou prozkoumány OGC standardy a specifikace, které se prolínají všemi výše zmíněnými úkoly a jsou nedílnou součástí celého řešení. Celé řešení by mělo v co největší míře tyto standardy splňovat a dodržovat.

Nad zvolenými technologiemi bude naprogramována aplikace pro automatickou publikaci metadat a dat dle požadavků výše. Protože část aplikací, které připadají v úvahu, jsou napsány v jazyce Java (např. GeoServer, GeoNetwork OpenSource), bude i tato aplikace vyvíjena v Javě. Aplikace bude zastřešovat veškeré operace zmíněné v této kapitole.

Celý koncept bude pojat jako možná budoucí součást SDI a bude proto nahlíženo k celkové koncepci aplikace tak, aby byla jednoduše převzatelná a implementovatelná do dalšího kódu.

Dosažené výsledky a aplikace budou testovány nad reálnými daty v prostředí GeoNetwork opensource, kde budou dostupná publikovaná data včetně metadat a mapového okna OpenLayers, kde bude možné ověřovat správnosti mapových výstupů a rychlost jejich načítání. Budou také ukázány možnosti připojení publikovaných vrstev do různých GIS aplikací.

KAPITOLA 2

Dostupné technologie, standardy a jejich zhodnocení

Vzhledem k zadání práce bylo nutné prostudovat možnosti technologií, které by měly být v práci použity. Požadavky na výslednou aplikaci a její předpokládané nasazení v rámci katederní SDI jasně eliminovaly možnosti, kterými se lze při výběru vhodného softwaru řídit.

Obecné požadavky na použitý software se nechají shrnout následovně. Veškerý použitý software by měl být volně dostupný. Open source a free software zaznamenal v poslední dekádě obrovský rozvoj a otevřené programy se mohou v mnoha ohledech plně poměřovat s komerčním software (Chen ... [et al.], 2010).

Použité programy by měly být platformě nezávislé. V případě, že by došlo k nasazení aplikace na jiné systémové prostředí, mělo by být možné vlastní program i programové součásti bez větších zásahů přenést.

GIS aplikace, které budou používány, musí splňovat OGC standardy a specifikace, aby nedocházelo ke komplikacím s využitím nestandardních dat v dalších GIS aplikacích, kde by mohla být data využívána.

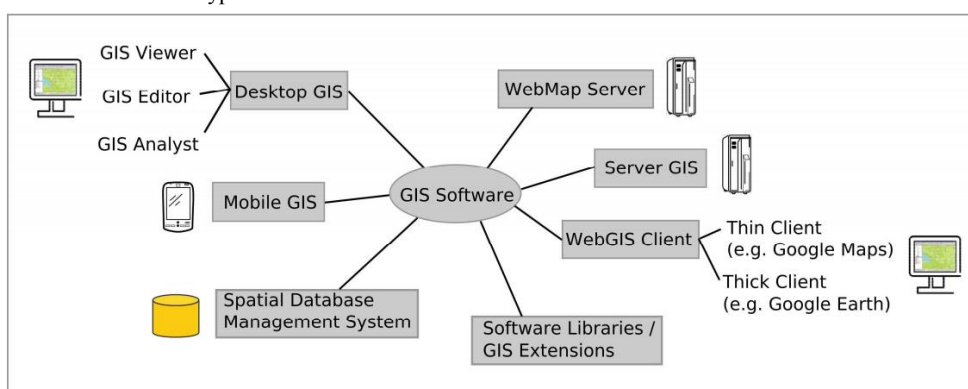
Samotné vývojové prostředí pro aplikaci by mělo být multiplatformní, aby bylo možné aplikaci sestavovat a ladit nezávisle na platformě, na které bude aplikace nasazena.

Vytvoření celé aplikace vyžaduje značné znalosti na poli webových GIS technologií, OGC standardů, prostorových databází a zároveň i přehled softwaru jak otevřených, tak komerčních a to jak po stránce uživatelské, tak po stránce vývojářské.

Tuto kapitolu můžeme rozdělit do tří bloků. První částí je řešerše OGC a ISO standardů, které budou nebo mohou být použity. Druhou částí je nastínění problematiky open source a svobodného software v GIS. Část třetí rozebírá problematiku dostupných technologií a je rozdělena do několika podčástí podle operací, které daná část vykonává v celkovém návrhu řešení. Velmi názorné dělení podle typu úloh, které daná část SDI poskytuje, může vypadat následovně (Steineger; Weibel, 2010).

- **Desktopový GIS** – tvorba, editace, analýzy dat a tvorba map
- **SDBMS** – datové sklady a v poslední době také analýzy prostorových dat
- **Mapové servery** – distribuce mapových podkladů po internetu
- **Serverové GIS** – analýzy dat a webový GIS processing
- **Weboví GIS klienti** – zobrazování a dotazování dat poskytovaných vzdáleně přes internet či intranet
- **Mobilní GIS** – nejčastěji používán pro sběr dat v terénu
- **GIS knihovny a rozšíření** – přidávají nové funkce a možnosti do všech předešlých kategorií

Obrázek 1: Přehled typů GIS



Zdroj: Steineger; Weibel, 2010

Dělení podle (Steineger; Weibel, 2010) popisuje obecně veškerý GIS software, avšak ne všechny části zde popisované budou v této práci využívány. Nebudou využity mobilní GIS a serverové GIS zpracovávající online analýzy.

2.1. OGC a ISO

Protože dochází ke stále většímu posunu od desktopových GIS směrem k distribuovaným verzím GIS řešení, je nutné se také věnovat otázce tzv. interoperability v GIS aplikacích. Aby zůstala interoperabilita v GIS zachována, vznikají standardy a specifikace (Sklenička, 2006).

OGC (Open Geospatial Consortium) je největší organizací zabývající se standardizací na poli geografických informačních systémů. OGC bylo založeno v roce 1994 jako mezinárodní průmyslové konsorcium a dnes má 467 členů (08/2012). Členskou základnu tvoří jak firmy, vládní agentury, tak například také univerzity (OGC, c1994-2012).

Vedle samotného konsorcia OGC je třeba zmínit organizaci W3C, která sice přímo nesouvisí s geoinformačními systémy, ale má největší podíl na standardizaci webového prostředí, na které jsou GIS technologie v prostředí webu navázány.

Dalším velkým hráčem mezi standardizačními orgány je ISO (International Organization for Standardization), která je tvůrcem mnoha mezinárodních ISO norem, které jsou známé i mimo

oblast GIS. Kupříkladu OGC WMS standard je zároveň ISO normou 19128 - Geographic information - Web map server interface (Steineger; Hunter, 2009).

Nakonec nesmíme opomenout také evropskou organizaci INSPIRE (The INfrastructure for SPatial InfoRmation in Europe).

Standardů OGC a ISO je nepřeberné množství, proto zde budou zmíněny pouze ty, které s prací souvisí a mohou být použity.

2.1.1. OWS

GIS se stále více přesouvá do sféry webového prostředí a vzniká stále více webových GIS standardů, proto je vhodné, aby tyto standardy měly nějaký společný rámec. Proto vznikla specifikace OWS (OpenGIS Web Services Common Specification), která tvoří právě toto společné začlenění.

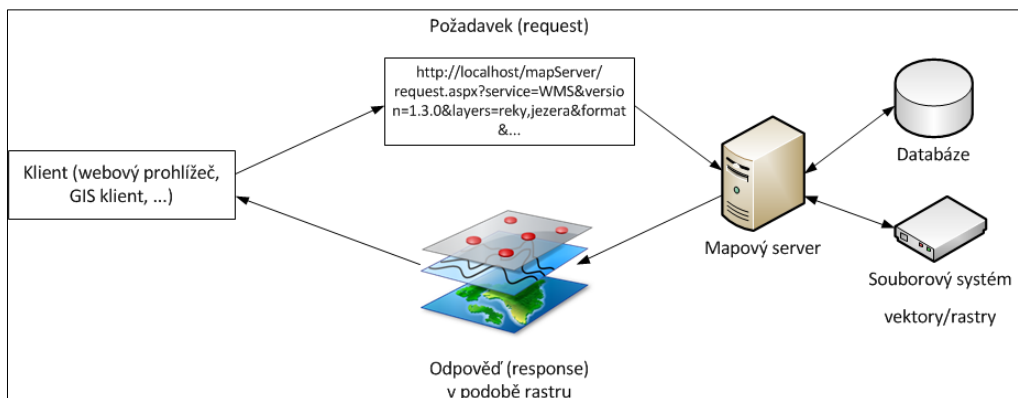
OWS definuje několik společných vlastností webových služeb. Jde o rámec určitých parametrů a obsahu klientských požadavků (např. všechny OWS služby mají operaci *GetCapabilities* – WMS, WFS, WCS, WMTS a další) a datových struktur, které služba vrátí (požadavky a odpovědi) (Sklenička, 2006).

Vzhledem k faktu, že OWS zastřešuje výhradně webové služby, komunikace mezi klientem a serverem probíhá pomocí HTTP protokolu.

2.1.1.1. WMS

Web Map Service je dnes nejpoužívanější a také nejjednodušší mapová služba. Mapový server zpracuje požadavek klienta, například webového prohlížeče, a odešle mu zpět georeferencovaný obrázek. Obvykle umí mapové servery vracet odpověď v různých formátech (JPEG, PNG, atd.). Pro upřesnění, mapový server je schopen zpracovávat jak vektorové, tak rastrové podklady, ale na klienta už se data dostanou pouze jako rastr.

Obrázek 2: Schéma požadavku GetMap WMS



Zdroj: vlastní

WMS má 3 základní operace:

- *GetCapabilities* – vrací informace o dané službě (název, kontaktní informace, seznam vrstev, měřítková omezení, atd.)

- *GetMap* – vrací samotnou mapu (v rastrové podobě)
- *GetFeatureInfo* – vrací informace o prvku (atributy prvku)

Každá z operací má povinné a nepovinné parametry. *GetCapabilities* má například povinné parametry *service* a *version*. Vzhledem k tomu, že již v požadavku *GetCapabilities* se vyskytuje parametr *version*, vyplývá z toho, že v dalších operacích se parametr *version* také projeví a to tak, že různé verze WMS mají např. různé názvy parametrů. Proto zde nebudeme vypisovat jednotlivé povinné a nepovinné atributy požadavků a spokojíme se se základními operacemi.

2.1.1.2. WFS

WFS (Web Feature Service) je velmi obdobnou službou jako WMS, ovšem pokud bychom rozdělili služby na vektorové a rastrové, tak WMS by spadala pod služby rastrové (na klientskou aplikaci se vrátí rastr) a WFS pod služby vektorové. Na klientskou aplikaci se vrátí GML – Geography Markup Language (Vohnout, 2009).

Standard WFS umožňuje díky rozšíření WFS-T (Transactional) editaci dat na klientovi a následné odeslání aktualizovaných dat zpět na server.

Se službou WFS umí pracovat většina GIS klientů, ať již v základní podobě nebo pomocí rozšíření, ovšem podpora není tak rozšířená jako u WMS.

V současnosti je dostupná verze WFS 2.0 (ISO 19142 Geographic information - Web Feature Service).

Základní operace WFS (bez transakcí) jsou:

- *GetCapabilities* – vrací informace o dané službě, název, kontaktní informace, seznam vrstev, měřítková omezení, atd.
- *DescribeFeatureType* – slouží ke zjištění příslušného XML schématu, aby klient správně rozpoznal jednotlivé tagy ve výsledném GML
- *GetFeature* – vrací samotná data ve formátu GML

2.1.1.3. WMTS

Web Map Tile Service je OGC standard sloužící k distribuci mapových dlaždic. Také patří do skupiny rastrových mapových služeb. Na rozdíl od WMS nejsou generovány rastry pro klienta při každém požadavku, ale vytvoří se dlaždicová síť rastrů („tilů“) v různých měřítkách, která je uložena na disku kam vidí mapový server, a tyto dlaždice jsou distribuovány mapovým serverem. Generování dlaždic do značné míry záleží na implementaci WMTS standardu (ústní informace od implementátorů WMTS na ČÚZK).

Z tohoto přístupu plyne několik výhod, ale také nevýhod.

Výhody:

- služba je o poznání rychlejší než běžné WMS a to i mnohonásobně
- lze „cachovat“ mapové tily na různých úrovních server, klient – dochází k překreslení části mapy a ne celé mapy jako u WMS, kde je mapa pokryta jedním požadavkem

Nevýhody:

- dlaždice musí být vygenerovány ve všech dostupných měřítkách a souř. systémech – z toho plynou velké objemy uložených dat – rozdílný přístup výrobců software, WMTS standard neurčuje, zda dlaždice musí být předgenerované či se generují dynamicky, většina implementací je řešena právě předgenerovanými dlaždicemi
- vzhledem k tomu, že jsou generovány dlaždice pro určitá měřítka, je nutné na klientské straně zajistit skokový posun měřítek a nikoliv kontinuální jako tomu může být u WMS
- data jsou ve většině implementací předgenerována tzn., že jakékoliv změny v databázi nebo zdrojových datech, ze kterých dlaždice vznikly, se neprojeví, dokud se dlaždice nebo jejich část nevygenerují znovu, z toho plyne, že WMTS služby nejsou vhodné pro často aktualizovaná data
- WMTS podporuje jen velmi málo desktopových GIS

Přes tyto nevýhody je WMTS velmi často používaná právě kvůli rychlosti. Samozřejmostí je jistá typologie pro data, se kterými je výhodné WMTS použít. Velmi také záleží na implementaci WMTS standardu.

Základní operace jsou:

- *GetCapabilities* – stejné jako u ostatních OWS služeb
- *GetTile* – vrací požadované dlaždice
- *GetFeatureInfo* – informace o vybraném prvku

2.1.1.4. CSW a ISO standardy metadat

Catalogue Services for Web je standardem OGC, který je používán pro komunikaci mezi metadatovými řešeními. Aby bylo možné data shromažďovat na jednom místě a umožnit tak vyhledávání napříč katalogy, využívá se služba CSW.

CSW umožňuje jak čtení v ostatních katalozích, tak i zápis metadat. Podporuje mnoho formátů metadat. Obecně jsou pro geoinformatiku a prostorová metadata nejzajímavější 3 ISO normy metadat a to sice ISO 19115, 19119 a 19139 (Vohnout, 2009).

Nejvíce používanou normou metadat je norma ISO 19115 (Geographic information – Metadata). Tato norma spadá do balíku norem ISO 191xx TC 211, které slouží k popisu prostorových dat, tato norma byla přijata v roce 2004 za českou normu. Ovšem norma sama nedefinuje společný formát metadat a v řadě aplikací dochází k rozdílnostem v implementaci XML formátu, proto vyšla v roce 2007 norma ISO 19139, která upravuje pravidla pro ukládání metadat do XML, tedy definuje závaznou strukturu metadatových XML dokumentů (Kafka, 2008).

Pro popis služeb, zejména těch, které spadají pod OGC standardy, slouží metadatová norma ISO 19119, která rozšiřuje ISO 19115.

2.1.2. WMC

Web Map Context definuje výměnný formát nezávislý na klientské i serverové straně, který obsahuje definici konkrétního seskupení jedné nebo více mapových služeb z jednoho nebo více mapových serverů.

WMC zahrnuje informace o serverech poskytujících vrstvy, mapové výřezy, definici souřadnicového systému, atd. Dokument slouží k ukládání a přenosu těchto informací mezi různými typy klientů i serverů. Přenosovým formátem je XML.

WMC může být například využito k nastavení výchozího stavu klienta pro skupinu uživatelů, ukládání stavu klienta (zobrazené mapy, měřítko, pořadí, souřadnicový systém).

2.1.3. SFS

Simple Feature Specification určuje způsob popisu základních grafických objektů, které se v GIS vyskytují (body, linie, polygony, ...). Dále také popisuje základní vztahy mezi nimi, jako například průsečík, překrytí a další. SFS existuje ve třech specifikacích OLE/COM, CORBA a SQL (Sklenička, 2006). Právě specifikaci SFS for SQL implementuje databáze PostgreSQL s prostorovou nadstavbou PostGIS.

2.1.4. KML

KML (Keyhole Markup Language) je rozšířením XML, sloužící pro práci s prostorovými daty. Původně byl formát vyvíjen firmou Keyhole, Inc., v roce 2004 koupil formát Google pro využití ve svých aplikacích Google Earth a Google Maps a v roce 2008 se stal KML standardem OGC. S přechodem formátu pod OGC se rozmáhá i využití formátu v GIS technologiích.

KML umožňuje uložení jak jednoduchých geometrií v podobě například jednoduché bodové geometrie, až po složité 3D útvary s texturami. U složitých útvarů je častý problém s velikostí souborů, proto se můžeme setkat s komprimovanou formou KML, která se jmenuje KMZ (Vohnout, 2009).

2.1.5. GML

Jazyk pro modelování, přenos a ukládání prostorových dat, včetně jejich prostorových i atributových informací. Vytvořen pomocí jazyka XML.

Jazyk GML využívají například firmy ESRI, Intergraph, Autodesk, MapInfo, Cadcorp. ČÚZK využívá tento jazyk jako jeden z formátů pro distribuci ZABAGED.

Aktuální verze jazyka GML je 3.2.1 - Geography Markup Language (GML) Encoding Standard a je přijat i jako ISO standard ISO 19136 – GML.

Hlavní rozdíl mezi GML a KML je v tom, že KML slouží primárně jako jazyk pro reprezentaci a stylování objektů, kdežto GML je jazyk pro uchování prostorových objektů a jejich atributů, nikoliv jejich grafické reprezentace.

2.2. Open source a svobodný software

Open source software je velmi jednoduše řečeno softwarem, u kterého jsou dostupné zdrojové kódy daného programu a zároveň je umožněna další práce s nimi a jejich šíření.

Přesnou definici jaký software může být považován za open source definuje Open Source Initiative (OSI). Open Source software podle OSI musí splňovat následující podmínky (zpracováno podle Aujezdský, 2009).

- Volné rozšiřování – licence nesmí omezovat prodej nebo jinou distribuci a to ani jako součást jiného programu.
- Zdrojový kód – musí být poskytnut zdrojový kód a to v takové formě, aby byl uživatel schopen kód modifikovat. Kód nesmí být záměrně znečitelnován. Musí být poskytnut jak zdrojový kód, tak i binární spustitelné soubory, a to zcela zdarma nebo s jasně definovaným postupem kde zdrojové kódy získat.
- Integrita zdrojového kódu – licence může omezit distribuci změněné formy zdrojového kódu pouze pokud jsou společně s kódem distribuovány updaty. Licence pak musí výslovně povolit šíření změněného zdrojového kódu a zároveň může požadovat, aby odvozené práce nesly jméno nebo verzi odlišné od původního programu.
- Diskriminace osob a skupin – licence nesmí diskriminovat osoby nebo skupiny osob.
- Diskriminace oblastí použití – licence nesmí omezovat použití v různých sférách jako např. komerční sféra, výzkum, atd.
- Aplikovatelnost licenčního ujednání – pro užití počítačového programu nesmí být uzavírána žádná další smlouva či dodatek.
- Licence nesmí záviset na programovém balíku – licence k programu nesmí být závislá na formě distribuce. Pokud je program distribuován jako součást balíku a následně je vyřazen a distribuován samostatně v souladu s licencí, měli by uživatelé, kteří program dostanou samostatně, mít stejná práva jako ti, kteří dostali program jako součást programového balíku.
- Licence nesmí ovlivňovat jiný software – licence např. nesmí stanovit, že všechny programy distribuované na jednom mediu se musí řídit podmínkami Open Source software.
- Licence musí být nezávislá na určité technologii.

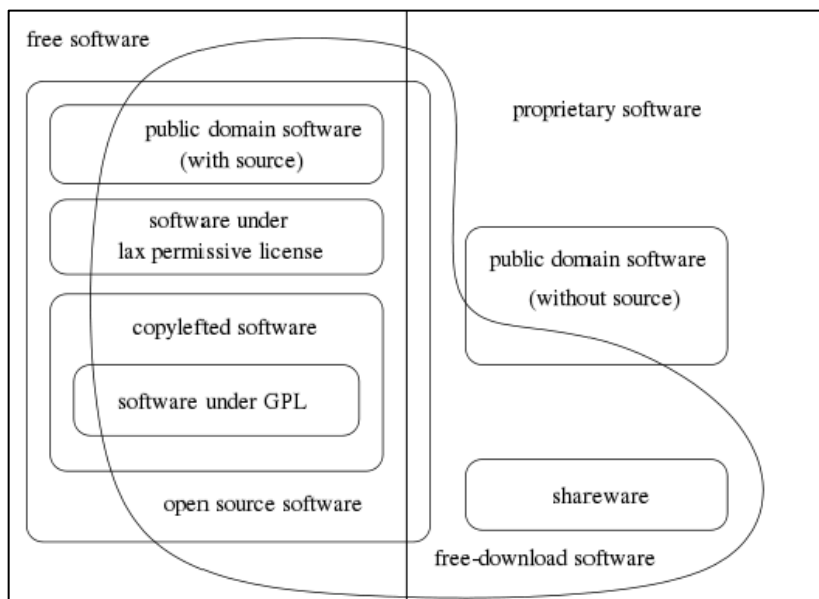
Často dochází k záměnám či rozporům mezi open source softwarem zaštiťovaným Open Source Initiative a free (svobodným) softwarem zaštiťovaným Free Software Foundation (FSF). Zásadní rozdíly mezi těmito dvěma pojmy neexistují. Hlavním společným rysem obou nadací je shoda v hlavní myšlence a to sice, že distribuce softwaru musí probíhat takovým způsobem, že uživatel má vždy právo na získání zdrojového kódu a dále může zdrojový kód modifikovat a dále distribuovat.

Problémem je použití výrazu free software, kde výraz „free“ dává jakýsi pocit, že by software měl být distribuován zdarma, což ale u svobodného software není nutnou podmínkou. Open source software je možné prodávat, ale opět pouze s podmínkou distribuce zdrojového kódu. Obecně se dává přednost používání pojmu open source před pojmem free software (OSS Alliance, 2012). Rozložení různých typů software zobrazuje obrázek 3.

Svobodný software by měl splňovat následující základní uživatelská práva (GNU Operating System, c1996-1999)

- Svoboda spustit program za jakýmkoliv účelem (svoboda 0).
- Svoboda studovat, jak program pracuje a přizpůsobit ho svým potřebám (svoboda 1). Předpokladem k výše uvedenému je přístup ke zdrojovému kódu.
- Svoboda redistribuovat kopie, abyste pomohli vašemu kolegovi (svoboda 2).
- Svoboda vylepšovat program a zveřejňovat zlepšení, aby z nich mohla mít prospěch celá komunita (svoboda 3). Předpokladem k výše uvedenému je přístup ke zdrojovému kódu.

Obrázek 3: Vztah kategorie software podle licencí



Zdroj: <http://www.gnu.org/philosophy/category.png>

2.2.1. Nejrozšířenější open source licence

Protože popis jednotlivých licencí zaštiťovaných OSI by vydal na vlastní práci, uvádím pouze přehled nejrozšířenějších licencí podle Open Source Initiative. Detailní popisy v původním znění je možné získat na stránkách OSI (<http://www.opensource.org>) nebo lze některé licence včetně komentářů v českém jazyce najít v (Aujezdský, 2009).

- Apache License, 2.0 (Apache-2.0)
- BSD 3-Clause "New" or "Revised" license (BSD-3-Clause)
- BSD 3-Clause "Simplified" or "FreeBSD" license (BSD-2-Clause)
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license (MIT)
- Mozilla Public License 2.0 (MPL-2.0)
- Common Development and Distribution License (CDDL-1.0)
- Eclipse Public License (EPL-1.0)

(zpracováno podle Open Source Initiative, 2012)

2.3. Open source GIS

2.3.1. Open Source Geospatial Foundation

Posledních šest let zastřešuje open source projekty spojené s GIS Open Source Geospatial Foundation (OSGeo). Jedná se o neziskovou organizaci, jejímž cílem je podporovat a prosazovat společný vývoj open source software na poli geoinformačních technologií, šířit open source software a zároveň sdružovat jak vývojáře, přispěvatele a další zdroje se záměrem použití těchto zdrojů ve prospěch veřejnosti. OSGeo zároveň plní funkci právního zázemí pro vlastní komunitu (OSGeo, 2012).

OSGeo dělí aplikace, které spravuje do čtyř skupin podle zaměření aplikace.

- **Webové aplikace**
Deegree, GeoMajas, GeoMoose*, GeoServer*, MapBender, MapBuilder, MapFish, MapGuide Open Source, MapServer, OpenLayers, Zoo*
- **Desktopové aplikace**
GRASS GIS, Quantum GIS, gvSIG, Opticks
- **Knihovny pro práci s prostorovými daty**
FDO, GDAL/OGR, GEOS, GeoTools, MetaCRS*, OSSIM, PostGIS, RasDaMan*
- **Metadatové katalogy**
GeoNetwork opensource

* aplikace jsou v „inkubačním stádiu“ OSGeo – aplikace ještě není zcela přijata jako open source a je ve stádiu přípravy pro přijetí do OSGeo (OSGeo, 2012).

OSGeo v rámci šíření open source ideí podporuje i tzv. *local chapters* – lokální jednotky OSGeo sdružující podobně smýšlející jedince a lokálně šířící myšlenky a programy OSGeo. Česká republika má vlastní *local chapter* (http://grass.fsv.cvut.cz/gwiki/OSGeo_Local_Chapter) sdružující se okolo české skupiny GRASS GIS.

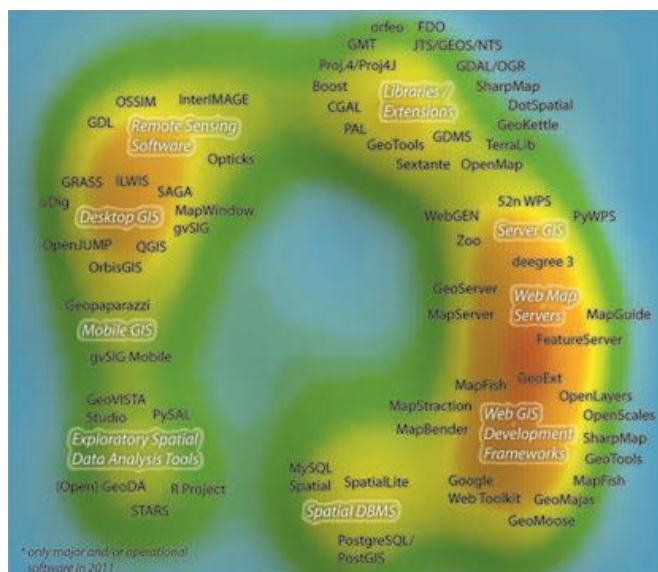
Za zmínku určitě stojí projekt OSGeo Live, který nabízí zdarma DVD s live distribucí operačního systému Xubuntu a předinstalovanými aplikacemi nejen z projektu OSGeo. Výhodou této distribuce je, že začínající uživatel nemusí procházet mnohdy složité instalační postupy a může začít rovnou pracovat, případně testovat různé aplikace a posléze se rozhodnout, zda je pro jeho účely aplikace využitelná. OSGeo Live obsahuje i ukázková data.

2.3.2. FOSS4G software list

Vedle oficiálních zdrojů open source softwaru existují i menší projekty, které jsou ale obrovským přínosem právě z důvodu sumarizace projektů nezávisle na jejich správci či vlastníkovi. Velmi častým neduhem těchto projektů je jejich neaktuálnost, která při rychlosti vývoje open source aplikací znamená fatální nedostatek, což ale neplatí pro projekt *FOSS4G software list*. K tomuto projektu a jeho autorům se také vážou články, které specifikují možnosti použití open source softwaru pro různé aplikace např. (Steineger; Hunter, 2009) nebo (Steineger; Bocher, 2009).

S tímto projektem je také spjat obrázek 4, který zobrazuje rozložení FOSS4G projektů napříč GIS spektrem různě zaměřených programových skupin. Je zde vidět, že se na obrázku nacházejí jak projekty OSGeo, tak i mimo tuto organizaci.

Obrázek 4: Výběr z FOSS4G projektů v roce 2011



Zdroj: http://sourceforge.net/userapps/mediawiki/mentaer/index.php?title=File:Projectsmap_v1_150dpi_grey.jpg

2.3.3. Ostatní zdroje open source a svobodného GIS

Často uváděnými zdroji open source GIS jsou webové stránky <http://www.opensourcegis.org/> a <http://www.freegis.org/>, první zdroj trpí již zmiňovaným problémem neaktuálnosti (poslední aktualizace ke dni 19.8.2012 byla z 12.12.2008), druhý zdroj je databází open source a free software projektů včetně dostupných volně stažitelných dat. Autorem druhého zdroje je německá oficiální OSGeo *local chapter* FOSSGIS e. V.

2.4. Proprietární GIS

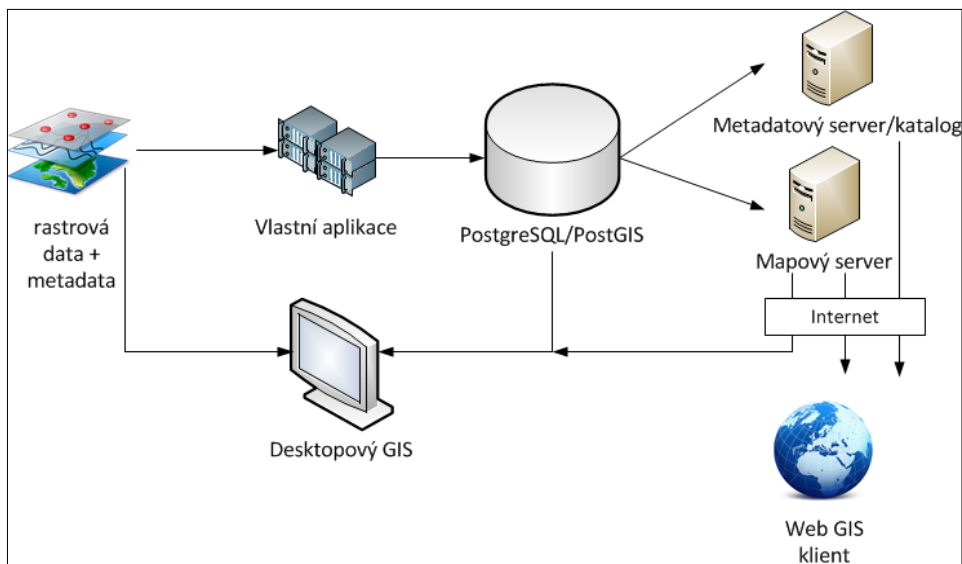
Nebylo by správné největší kategorii GIS technologií úplně vynechat, ale vzhledem k předpokládanému nasazení výsledné aplikace v akademické sféře a primárních požadavcích na svobodný případně open source software, bude tato kategorie zmiňována v této práci pouze okrajově.

2.5. Komponenty SDI pro automatickou publikaci dat a metadat z PostgreSQL

Bylo nutné specifikovat, jaké komponenty (části) SDI budou potřeba k naplnění požadavků na výslednou aplikaci. Aplikace bude vyžadovat následující součásti, které jsou vymezeny podle (Steineger; Weibel, 2010).

- SDBMS
- GIS knihovny a rozšíření
- Desktopové GIS
- Mapové servery
- Weboví GIS klienti
- Mimo toto členění stojí **metadatové katalogy**

Obrázek 5: Obecný koncept fungování aplikace MTDTRasPub



Zdroj: vlastní

V následujících kapitolách jsou popsány aplikace splňující obecné požadavky definované na začátku kapitoly 2 nebo ty aplikace, které by mohly adekvátně nahradit některou z částí systémů tak, aby byla zachována funkcionality systému. V případě, že by byla dostupná z komerčních technologií, budou popsány i některé komerční možnosti.

2.5.1. Prostorové databázové systémy s podporou rastrových dat (SDBMS)

Prostorové databáze jsou velmi zjednodušeně řečeno databáze, které podporují práci s prostorovými objekty. K prostorovým objektům se databáze chová stejně jako k ostatním datovým typům v databázi (Blasby, 2001).

Pokud by se práce měla zabývat všemi databázemi včetně těch komerčních s podporou prostorových objektů, velká část by musela být věnována pouze tomuto úkolu. Vzhledem k cíli práce jako open source řešení, budou pouze porovnány nejpoužívanější komerční databáze Oracle s prostorovou nadstavbou Oracle Spatial s open source databázovými možnostmi, které by připadaly pro práci v úvahu.

Pro přehled jen uvádím výčet komerčních databází s podporou prostorových dat (nikoliv přímé uložení rastrových dat). Příklady databází s podporou prostorových dat např. Boeing's Spatial Query Server, Smallworld VMDS, IBM DB2 Spatial Extender, Oracle Spatial, Microsoft SQL Server, atd. (Wikipedia, Spatial database, 2012).

Prostorové databáze nejsou obvykle vyvíjeny jako primárně prostorové, ale prostorová složka je přidávána do již existujících databází jako jejich rozšíření (Steineger; Hunter, 2012).

Pravděpodobně nejpoužívanější open source prostorovou databází s podporou rastrů je PostgreSQL s prostorovou nadstavbou PostGIS.

2.5.1.1. PostgreSQL/PostGIS/PostGIS raster

PostgreSQL se svojí nadstavbou PostGIS je jednou z nejrozšířenějších prostorových open source databází. PostGIS raster (dříve známý také jako WKT Raster) rozšiřuje tuto databázi o možnost ukládání a práce s rastry. O rozšíření PostGIS svědčí podpora v mnoha open source i komerčních GIS projektech. To je také jedním z důvodů, proč je PostgreSQL základním stavebním kamenem této práce. Pokud si stanovíme požadavky tak, že databázová platforma musí umět ukládat a manipulovat s rastry, ať již uvnitř databáze nebo pouze jako odkazy rastrů uložených v souborovém systému, ukládat vektorová data, dále komunikovat jak s desktopovými GIS, tak i webovými aplikacemi, možnosti výběru se velmi zúží.

Právě PostgreSQL je databáze, která tyto podmínky splňuje jako jediná z open source databází.

Tabulka 1: Požadavky na databázi PostgreSQL

Vlastnost	Ano/Ne	Poznámka
Open Source	✓	
Manipulace s vektorovými daty	✓	
Manipulace s rastrovými daty	✓	
Připojení k desktopovým GIS	✓	Rastry zatím pouze QGIS a gvSIG (08/2012), vektory velmi rozšířené i do komerční sféry – ArcGIS, GeoMedia, ...
Připojení k webovým GIS	✓	Rastry pouze UMN Mapserver a GeoServer

Zdroj: vlastní

2.5.1.2. SQLite/SpatialLite/RasterLite

Databáze SQLite se svoji prostorovou nadstavbou SpatialLite a podporou rastrů RasterLite sice také splňuje většinu požadavků, bohužel zatím není možné připojit rastrové podklady do desktopových a webových GIS, proto není možné použití databáze SQLite pro účely této práce.

SQLite je databáze, která je tzv. „bezserverová“ – databáze běží, aniž by potřebovala ke svému běhu serverovou platformu. Aplikace komunikují napřímo se souborem databáze v souborovém systému (celá databáze je uložena včetně triggerů, pohledů, funkcí a dalších databázových součástí v jednom souboru). Databáze vyžaduje jen minimum závislostí na systému, na kterém běží a dalších externích knihovnách. Zároveň vyžaduje minimální konfiguraci. Toto předurčuje databázi k použití na mobilních zařízeních a jako lokální databáze namísto např. komerční MS Access, zvláště uvažíme-li, že velikost knihoven se všemi funkcemi SQLite se pohybuje kolem 350 kB (SQLite, 2012).

Tabulka 2: Požadavky na databázi SQLite

Vlastnost	Ano/Ne	Poznámka
Open Source	✓	
Manipulace s vektorovými daty	✓	
Manipulace s rastrovými daty	✓	
Připojení k desktopovým GIS	✓	Pouze nástroje LibreAtlas a spatialite-gis – žádné standardní GIS
Připojení k webovým GIS	✗	Zatím není známá možnost

Zdroj: vlastní

2.5.1.3. Oracle/Oracle Spatial

Oracle je jako nejrozšířenější komerční databázová platforma podporován i v řadách open source projektů. Jako jediný by bylo možné ho použít v rámci open source desktopových GIS, protože minimálně Quantum GIS má podporu jak vektorových vrstev uložených v Oraclu, tak zároveň i rastrových vrstev typu Oracle GeoRaster.

Ovšem Oracle má 2 odlišné licence týkající se prostorových dat, jejichž rozdíly jsou zásadní pro práci s rastry. První licencí je Oracle Locator, která je distribuována společně se všemi i volně dostupnými verzemi Oraclu včetně verze Express. Tato licence v sobě ovšem nezahrnuje podporu práce s rastry. Ta je podporována až v druhé licenci, která je ale distribuována pouze s Enterprise licencí Oracle, která je finančně nákladná (Oracle, 2012).

Využití databáze Oracle by připadalo v úvahu pouze, pokud by instituce disponovala Enterprise licencí Oracle. Na straně druhé, z pohledu aplikačního nasazení, by zde neměly být žádné zábrany vyjma finanční stránky, které by zamezovaly Oracle s nadstavbou Spatial použít.

Tabulka 3: Požadavky na databázi Oracle

Vlastnost	Ano/Ne	Poznámka
Open Source	✗	
Manipulace s vektorovými daty	✓	
Manipulace s rastrovými daty	✓	
Připojení k desktopovým GIS	✓	
Připojení k webovým GIS	✓	

Zdroj: vlastní

2.5.1.4. Ostatní SDBMS (bez podpory rastrových typů)

Ostatní open source databáze nemají podporu rastrových typů a proto se jimi práce nebude obsáhleji zabývat. Vzhledem k faktu, že do této kategorie spadá například nepoužívanější open source databáze MySQL a některé typově zajímavé databáze jako MongoDB či CouchDB, nebylo by správné tuto kategorii zcela přeskočit.

Některé tyto databáze mají ovšem určitá omezení např. MySQL Spatial Extension využívá při analýzách a dotazech pouze MBR (Minimum Bounding Rectangle) namísto vlastní geometrie, takže dochází ke značným zkreslením ve výsledcích. Mimo jiné zde za zmínku také stojí například databáze H2 nebo Ingres GeoSpatial (Steineger; Hunter, 2009).

Databáze jako například MongoDB či CouchDB nejsou klasickými relačními databázemi, ale jsou to tzv. dokumentové databáze nebo také NoSQL databáze. Tyto databáze mají podporu prostorových dat, avšak dotazování a práce s těmito databázemi je odlišná od klasických relačních databází podporujících SQL (Thon, 2010).

2.5.2. GIS knihovny a rozšíření

Protože knihoven pro práci s prostorovými daty je nepřeberné množství, budou zde zmíněny pouze ty, které budou pro naši práci potřebné. Do značné míry je výběr knihoven dán použitými technologiemi, které jsou součástí používaných aplikací.

Knihovny lze rozdělit do 3 skupin podle příslušnosti k programovacímu jazyku (platformě), ve kterém je knihovna napsána (Ramsey, 2007).

- Knihovny v jazyce C – GDAL/OGR, Proj.4, GEOS (C++ port JTS) + skriptovací jazyky jako Python, Perl nebo PHP
- Knihovny v jazyce Java – GeoTools, JTS (Java topology suite)
- Knihovny na platformě .NET – NTS (C# .NET port JTS), SharpMap

2.5.2.1. GDAL/OGR (Geospatial Data Abstraction Library)

GDAL/OGR je nejrozšířenější knihovna pro práci s rastrovými (GDAL) a vektorovými (OGR) daty.

Většina desktopových, ale i webových GIS v open source sféře podporuje práci buď s vektorovými nebo rastrovými formáty právě prostřednictvím knihovny GDAL/OGR. Opravdu

rozsáhlý seznam programů, které využívají některou ze součástí GDAL/OGR je k dispozici na webu GDAL/OGR. Namátkou z komerční sféry např. ESRI ArcGIS, FME, Oracle GeoRaster, Idrisi, Google Earth a ze sféry open source například GeoServer, MapServer, QGIS, uDIG, PostGIS a mnohé další (GDAL, 2012).

Mimo samotný přístup k datům GDAL/OGR disponuje také mnoha nástroji pro příkazovou řádku, které slouží pro transformace dat a různé manipulace s daty.

2.5.2.2. Proj.4

Proj.4 je knihovna umožňující přepočty mezi souřadnicovými systémy a řešící úlohy matematické kartografie. Jde především o úlohy přepočtu zeměpisných souřadnic ϕ , λ do rovinných X, Y a obráceně, výpočty zkreslení, atd. (Ježek, 2006).

Knihovnu Proj.4 využívá většina open source projektů pro transformaci mezi souřadnicovými systémy. Zároveň s knihovnou je k dispozici několik nástrojů pro příkazovou řádku.

2.5.2.3. GEOS (Geometry Engine, Open Source)

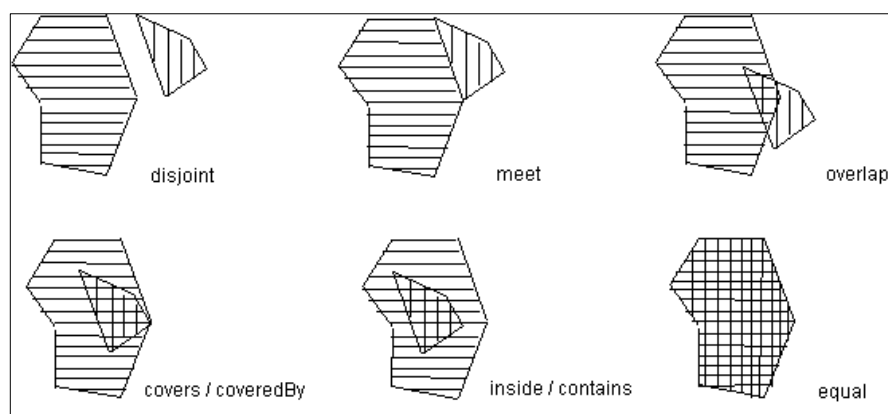
GEOS je implementací JTS pro jazyk C++. V této knihovně jsou k dispozici všechny objekty ze specifikace OGC Simple Feature Specification a zároveň všechny metody příslušné daným objektům (Ramsey, 2007).

Knihovnu GEOS využívá například PostGIS pro práci s prostorovými daty.

2.5.2.4. JTS (Java topology suite)

JTS je centrem vývoje GIS programů nejen v jazyce Java, ale jak již bylo uvedeno výše JTS má své implementace i v jazycích C# a C++. JTS striktně dodržuje standardy OGC Simple feature specification. Má velmi robustní implementaci prostorových predikátů (funkce, které porovnávají dva prostorové objekty a vracejí pravdu nebo nepravdu při splnění či nesplnění podmínek predikátu). V podstatě neexistuje případ, kdy by JTS nebyla schopná zpracovat výsledek predikátu nebo interpretovala výsledek chybně. Toto je obrovská výhoda JTS, kdy ani většina proprietárních aplikací nemá tyto funkce takto robustně vyřešeny (Ramsey, 2007).

JTS také implementuje prostorové operátory, jako je například rozdíl, sjednocení, průnik, atd., které byly důkladně testovány, ale nejsou tak robustní jako predikáty. Implementace prostorových predikátů a operátorů je velmi přínosná, protože je velmi obtížné tyto části naprogramovat s ošetřením všech možných případů, které mohou nastat. Proto je knihovna JTS velmi často používanou knihovnou OSS projektů (Ramsey, 2007).

Obrázek 6: Ukázky prostorových predikátů

Zdroj: http://jt.wz.cz/vytvory/spatial_sql/spatial_sql.htm

2.5.2.5. GeoTools

GeoTools je Java open source toolkit pro vývoj GIS aplikací podporující OGC a ISO standardy v co největší míře. GeoTools jsou velmi snadno modifikovatelné a lze velmi jednoduše přidávat, či odebírat jejich součásti a ty pak případně používat jako separátní knihovny (Ramsey, 2007), čehož bude využíváno v praktické části práce.

Cílem projektu GeoTools není vyvinout finální aplikaci, ale spíše poskytnout standardizované nástroje pro vývoj GIS aplikací. Knihovna je napsána modulárně tzn., že mohou být použity opravdu jen ty části, které potřebujeme k nějakému účelu, zatímco ostatní části mohou být vypuštěny. Moduly jsou většinou navrhovány tak, aby každý modul splňoval jeden OGC standard a podporoval interakci s různými datovými zdroji. Každý modul má svého vlastního vývojáře, který kontroluje obsah a směřování daného modulu (Ramsey, 2007).

Z obecného pohledu by se GeoTools nechaly srovnat komerčními ESRI ArcObjects nebo Intergraph GeoMedia Objects.

Tabulka 4: Použité GIS knihovny

Název	Implementační jazyk	Licence	URL	Odpovědnost za vývoj
GDAL/OGR	C++	MIT	http://www.gdal.org	OSGeo
Proj.4	C	MIT - style	http://trac.osgeo.org/proj/	OSGeo
GEOS	C++	LGPL	http://trac.osgeo.org/geos/	OSGeo
JTS	Java	LGPL	http://www.vividsolutions.com/jts/jtshome.htm	Vivid solutions
GeoTools	Java	LGPL	http://www.geotools.org/	OSGeo

Zdroj: Ramsey, 2007; vlastní šetření kvůli aktualizaci údajů

2.5.3. Desktopové GIS

Desktopové GIS projekty můžeme považovat za jakousi vstupní bránu k celé sféře GIS. Tito GIS klienti mohou být jak tlustí ve formě robustních analytických a komplexních nástrojů, tak tencí v podobě různých software, které slouží například k prohlížení a tisku dat. Z této kategorie GIS vznikaly postupně další dnes dostupné technologie. Rozvoj GIS začal nejdříve v armádě a poté na univerzitách. S rozvojem internetu dochází k přesunu GIS aplikací od desktopu k webovým řešením, která jsou dostupná mnohem většímu počtu uživatelů.

Desktopových GIS projektů splňujících většinu podmínek, které požaduje práce do výsledného konceptu aplikace, by bylo mnoho. Ovšem pokud je zohledněno poslední a zásadní kritérium, kterým je načítání a manipulace s rastry v PostgreSQL, rázem se nám výběr zúží na následující dva kandidáty, kterými jsou *Quantum GIS* a *gvSIG*.

Zatím není k dispozici jediná komerční aplikace, která by uměla přímo alespoň číst rastry uložené v databázi. Pokud bychom mohli použít databázi Oracle, tak datový typ Oracle GeoRaster je možné přečíst například jak v ArcSDE, tak i v open source Quantum GIS, takže z tohoto pohledu by mohla být považována databáze Oracle za výhodnější, s obrovským finančním zatížením na straně druhé.

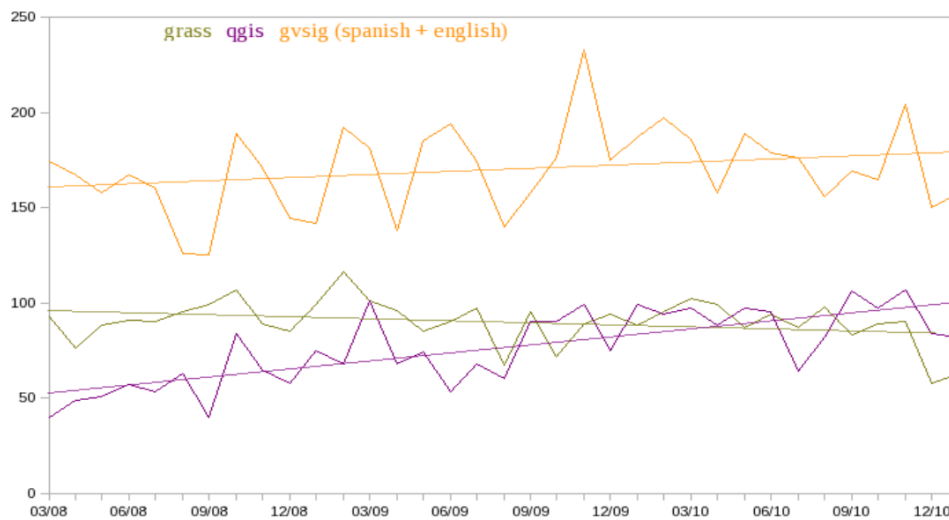
Protože návrh celého konceptu výsledné aplikace počítá s nasazením mapového serveru, bude možné vypublikovaná data připojit do desktopových klientů podporujících OGC WMS standard, případně WFS standard pro vektorová data.

Rozdíl mezi oběma GIS aplikacemi je pro naše účely zanedbatelný, obě aplikace splňují všechny požadavky, které jsou na desktopovou aplikaci v našem případě kladeny. Je tedy spíše na uživateli, pro kterou z aplikací se rozhodne a která případně bude lépe splňovat požadavky kladené uživatelem.

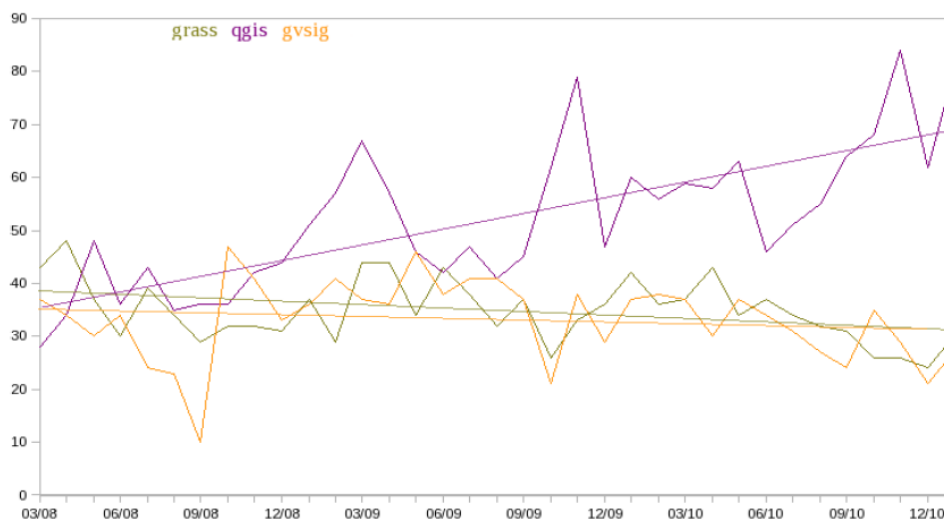
Pro tuto práci byla volba vcelku jasná s ohledem na předchozí zkušenosti s Quantum GIS, byl i pro tuto práci použit právě tento software.

Dalším kritériem při rozhodování, kterou z aplikací si vybrat, by mohly být např. dlouhodobé trendy vývoje aplikace, uživatelské a vývojářské komunity a dalších faktorů, které reflektuje (Maneiro ... [et. al], 2011). V tomto dokumentu je popsán stav open source projektů GRASS, gvSIG a Quantum GIS mezi léty 2008 – 2010. Obrázek 7 zobrazuje počet přispěvatelů do tzv. mailing listu jednotlivých aplikací, což je u open source projektů jedním z hlavních míst pro výměnu informací a podporu k jednotlivým projektům. Naopak obrázek 8 popisuje vývoj na straně vývojářské komunity, kam mají přístup pouze vývojáři podílející se na daném projektu.

Podobné analýzy mohou také výrazně pomoci při výběru open source projektů a napovědět, kam by se mohl daný projekt ubírat a jakou rychlostí. Takových průzkumů není mnoho k dispozici.

Obrázek 7: Počet uživatelů přispívajících do mailing listu (2008 - 2010)

Zdroj: (Maneiro ... [et. al], 2011)

Obrázek 8: Počet vývojářů přispívajících do dev-mailing listu (2008 - 2010)

Zdroj: (Maneiro ... [et. al], 2011)

2.5.4. Mapové servery

V kapitole 2.1.1 pojednala o mapových službách splňující standardy OGC, aby bylo možné je využívat, je nutné mít k dispozici software, který bude dané OGC služby a standardy implementovat. Takovým nástrojem jsou právě mapové servery.

U mapových serverů je tomu podobně jako u předchozí kategorie desktopových GIS a přicházejí zde v úvahu dvě možnosti. Těmi jsou dva největší mapové open source servery s podporou PostGIS raster a to sice **UMN Mapserver** a **GeoServer**. Žádný jiný produkt z kategorie webových mapových serverů zatím nepodporuje uložení rastrů v PostgreSQL.

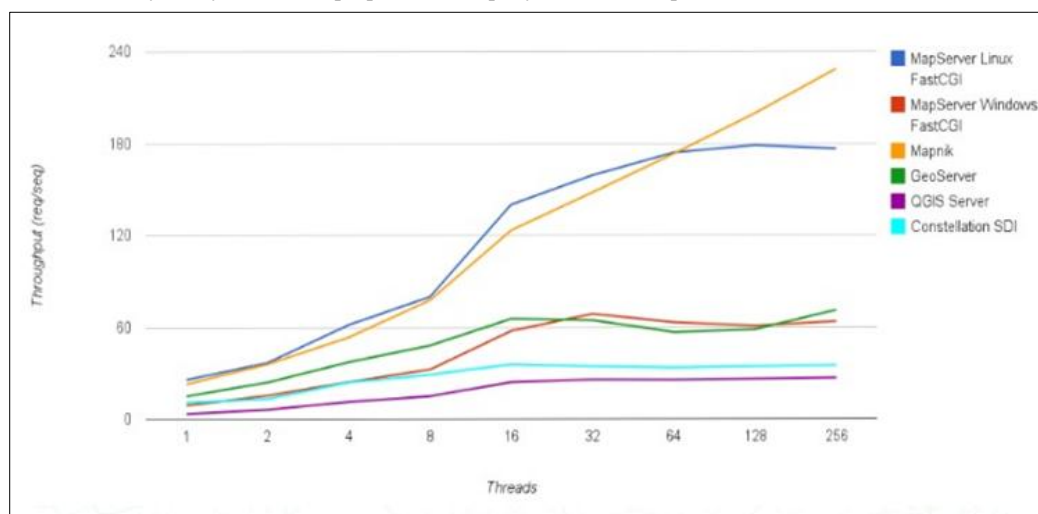
V této kategorii bylo rozhodování o využití technologie výrazně nejtěžší, protože oba mapové servery mají rozsáhlé a velmi podobné možnosti a zhodnotit, který je k danému účelu vhodnější byl nelehký úkol.

Pomineme-li základní vlastnosti obou mapových serverů a podíváme se na jejich zhodnocení z pohledu výkonu, byly na tom oba servery na platformě Windows velmi podobně (viz obrázek 9). Ovšem pokud se podíváme na výsledky MapServeru na platformě Linux jsou výsledky porovnání GeoServeru a MapServeru propastné, hlavně při větší zátěži (OSGeo, 2011).

Výsledky hodnocení je třeba opravdu porovnávat každý rok, kde je vidět, že náskok MapServeru se snižuje každým rokem. Je otázkou, zda nebude úzkým hrdlem GeoServeru platforma Java, která je obecně známá svou náročností na systémové prostředky.

Dalším porovnáním může být (Flower, 2011), které ale přináší velmi podobné výsledky, kdy v některých položkách vítězí MapServer, v jiných zase GeoServer.

Obrázek 9: Výsledky hodnocení propustnosti mapových serverů na počtu klientů



Zdroj: OSGeo WMS Benchmarking, 2011

Obecně by se nechalo stanovit, že co se rychlosti týče, bude MapServer o něco rychlejší, je třeba zamyslet se nad ostatními funkcemi, kterými je GeoServer o krok napřed.

- GeoServer má webové administrátorské prostředí.
- Disponuje REST API, přes které je možné celý mapový server ovládat, MapServer nic takového nemá.
- Jedna vypublikovaná vrstva se zpřístupní jako WMS, WFS, KML, atd., koncový uživatel má tedy více možností, jak službu zobrazit ve více klientech. V případě MapServeru platí co služba, to *mapfile* – konfigurační soubor mapové služby.
- GeoServer má základní správu uživatelů a rolí – není ji třeba řešit přes webový server.
- Umí běžet jako Windows služba, kde nemusí kolem sebe mít servletový kontejner v podobě webového serveru Apache Tomcat (jako webový server je použit Jetty).

Je otázkou priorit a zároveň také času, který chceme a můžeme věnovat vývoji aplikace, kterou potřebujeme nad mapovým serverem postavit. Jak říká Paul Ramsey: „MapServer je motor, dobrý motor. GeoServer je kompletní auto, ale zkuste s autem letět“. Zároveň je ale třeba dodat,

že motor bez kol nepojede a vymyslet a postavit zbytek auta nad dobrým motorem také není jednoduchou záležitostí. MapServer má větší potenciál pro nasazení na větších projektech, ale při správné konfiguraci by měl i GeoServer k většině operací naprosto postačovat.

Protože k využití Geoserveru nabádají i další okolnosti jako je znalost jazyka Java, nasazení GeoServeru společně s GeoNetwork opensource a předchozí zkušenosti se samotným GeoServerem na pokročilé úrovni, rozhodl jsem se i tentokrát použít jako mapový server GeoServer.

2.5.5. Weboví GIS klienti

Již byly popsány vybrané OGC webové standardy, mapové servery jako jejich produkční prostředí a desktopové GIS produkty jako konzumenti mapového obsahu. Aby byly popsány kompletní možnosti, je nutné zastavit se u kategorie, tzv. *web browser based clients* (Steineger, Hunter, 2009), která se rozšiřuje společně s přesunem GIS na internet.

V podstatě samotný prohlížeč může být webovým GIS klientem, ale pouze do jisté míry a značně nekomfortně. WMS je možné volat přímo z webového prohlížeče a prohlížeč obdrží stejný výsledek jako v některém z webových GIS klientů. Ovšem problémem je manipulace s mapou a uživatelský nekomfort, ale i takto lze pojmut webového GIS klienta.

Aby bylo možné s mapou v prostředí webových prohlížečů pracovat, vzniklo mnoho projektů, z nichž největší je projekt OpenLayers, který umožňuje připojení mnoha různých typů standardizovaných služeb. OpenLayers jsou postavené na technologii JavaScript. Dále je možné jmenovat například Adobe Flex prostředí OpenScales, projekty jako jsou například ka-Map, Chameleon a další. Pro úplnost je dobré ještě zmínit projekty jako SLMaPViewer, který je napsán v jazyce Silverlight a je prostředím pro uživatele UMN MapServeru (Steineger, Hunter, 2009).

Další kategorií, kterou je možné vyčlenit mezi webovými GIS klienty, jsou „portálová řešení“ jako například MapBender, GeoMajas, GeoMoose nebo GeoExt (Steineger, Hunter, 2009).

2.5.6. Metadatová řešení

Katalogové servery, nebo spíše raději metadatová řešení, která často zahrnují jak server, tak klientskou část, jsou posledním článkem potřebným k sestavení našeho konceptu.

Metadatové katalogy slouží k uchovávání, vyhledávání a manipulaci s metadatovými záznamy, tedy popisnými informacemi o datech.

Největším a také nejznámějším open source metadatovým řešením je projekt GeoNetwork opensource, který byl založen a je také využíván velkými organizacemi jako například Food and agriculture organization of the United Nations (FAO), World food programme (WFP) a dalšími (Ožana, 2007). Tento program bude také využíván pro naši práci.

Jako další možnosti z open source sféry můžeme jmenovat MDweb nebo například Deegree, které je kombinací jak mapového serveru, tak i metadatového řešení a webového frameworku pro práci s prostorovými daty (Steineger, Hunter, 2009).

Poslední aplikace CatMDEdit nespadá pod stejná řešení, která byla zmiňována v této kapitole, neboť je pouze desktopovým klientem (editorem) a nikoliv serverem poskytujícím metadata. O to větší má význam například při konverzích různých standardů metadat a poskytuje zajímavý nástroj pro práci a přípravu metadat, například pro import a export. V této práci byla použita pro konverzi metadat mezi standardem MARC21 (bibliografický formát metadat) a ISO 19115.

KAPITOLA 3

Koncept a struktura řešení

Před samotným sestavením řešení problému bylo nutné důkladně prostudovat vybrané technologie a seznámit se s jejich možnostmi, aby bylo možné navrhnout finální koncept aplikace.

Obecný návrh je postaven na standardech, které by bylo možné použít i mimo zvolené technologie. První částí je navržení obecného přístupu k metadatům a dalším nutným informacím, které je potřeba k publikovanému rastru znát. Zde byl zvolen přístup, který je dnes velmi často a efektivně využíván a který je jak strojově, tak lidsky snadno čitelný a to sice uložení doplňujících dat jako XML soubory. Tento princip počítá s uložením validních XML souborů s informacemi, které jsou nutné k vypublicování jak metadat, tak dat, které jsou na vstupu do aplikace (referenční soubory, soubory nutné pro mapové servery – EPSG kódy, názvy vrstev, případně další doplňující informace).

Dalším krokem je příprava metadat a dat pro publikaci a zároveň příprava databázových struktur. V tomto kroku je nutné také propojit data, která můžeme získat z rastrů pomocí dostupných technologií s metadaty a mohla by se objevit v metadatovém záznamu. Vhodnou cestou by v tomto případě bylo data s metadaty vůbec nespojovat a očekávat na vstupu celistvá a kompletní metadata a při publikaci doplnit do metadat pouze nové online zdroje, které vzniknou vypublicováním služeb v mapovém serveru. Potom bychom mohli zcela oddělit metadata od dat.

Poté co jsou připravená data a struktury, je potřeba data nahrát na odpovídající místa. Metadata a rastry do databáze. Informace nutné k vypublicování služeb a metadat na mapové a katalogové servery. Pokud se budeme držet standardů, u metadat by měla komunikace mezi aplikací a metadatovým katalogem probíhat přes službu OGC CSW.

Mnohem složitějším problémem, co se konceptu týká, je publikace samotných mapových služeb. Zde nelze stanovit obecný postup, jak data publikovat. Každá technologie vyžaduje vlastní přístup a neexistuje žádný standard, či specifikace, který by definoval, jak publikovat

data v mapovém serveru (jak založit novou mapovou službu). Proto k úkolu musí být přístupováno v závislosti na technologii a zvoleném datovém zdroji.

Poslední částí je publikace uživatelsky sestavených kompozic (definice více vrstev, měřítko, souřadnicové systémy, atd.), kde by bylo vhodné použít OGC standard WMC.

3.1. Možnosti uložení rastrů v PostgreSQL

Abychom pochopili mechanismus uložení rastrů v PostgreSQL, je třeba upřesnit, že databáze PostgreSQL s rozšířením PostGIS raster umožňuje 2 typy uložení rastrů, tzv. *in-database* a *out-of-database* (Obe; Hsu, 2011). Každý z těchto dvou typů má své výhody a nevýhody.

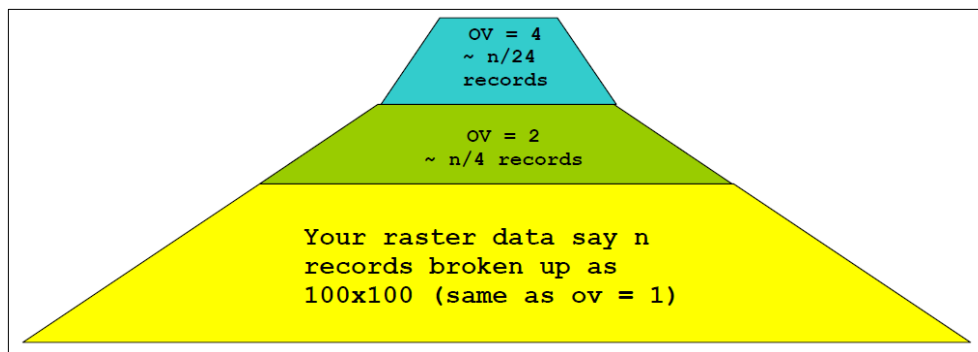
3.1.1. in-database

Pokud je použit typ uložení dat in-database, pak jsou uloženy samotné pixely uvnitř databáze v nativním datovém typu PostGIS raster, podobně jako jsou uloženy vektorové geometrie v datovém typu *geometry* nebo *geography*. Rastry mohou být uloženy ve dvou strukturách a to sice tak, že jedna tabulka odpovídá jednomu rastru (obsahuje jeden řádek s datovým typem raster) nebo druhou možností je uložení ve struktuře, kdy jedna tabulka obsahuje jednotlivé dlaždice rastru (rastr je rozdělen do více menších dlaždic) a tato tabulka dává dohromady jeden celek (Obe; Hsu, 2011).

Z našeho pohledu se jeví výhodnější použít systém, kdy rastry budou rozděleny do dlaždic, které je pro webové využití vhodné generovat v rozsahu od 200 x 200 pixelů do 400 x 400 pixelů a v případě využití rastrů pro analýzy od 50 x 50 pixelů do 200 x 200. Důvodem dělení rastrů na dlaždice je jednak jejich velikost v případě velkých rastrů a druhým důvodem je mnohem rychlejší manipulace s menšími částmi rastru. Datový typ raster, stejně jako geometry, využívá v PostGIS prostorového indexu GiST (Obe; Hsu, 2011).

Výhodou databázové nadstavby PostGIS je možnost importu rastrových dat včetně tzv. dlaždic a pyramid, které výrazně zrychlují zobrazování rastrů při posunu napříč různými měřítky (Obe; Hsu, 2011 [a]). V databázi jsou pyramidy uloženy jako další tabulky s rastry, které mají velikost pixelu úměrně zmenšenou hladině náhledu a mají vytvořeny dlaždice. Podobné řešení má například i Oracle Spatial (Racine, 2011).

Obrázek 10: Příklad pyramid nad rastrem



Zdroj: Obe, Hsu, 2011

Výhody:

- rastry mohou využívat obecně všech vlastností relačních databází
 - vzhledem k době, po kterou je PostGIS raster vyvíjen (cca 3 roky) je tento způsob uložení mnohem více otestovaný než druhý typ *out-of-database*
 - je zajištěna plná transakční integrita databáze při editaci rastrů
 - oproti *out-of-database* přístupu je tento přístup rychlejší ve čtení dat a různých analýzách
 - rastry mohou být zálohovány společně s databází (jsou jejím vnitřním typem)
- (Obe; Hsu, 2011)

Nevýhody:

- rastry jsou obecně větší než vektorová data, proto mohou zálohy databáze zabrat s rastry uvnitř databáze mnoho času, proto je nutné dobře navrhnout strukturu databáze, aby byla oddělená často aktualizovaná data a od méně často aktualizovaných
 - není možné jednoduše pracovat s rastry v databázi v aplikacích, které je nepodporují
- (Obe; Hsu, 2011)

3.1.2. out-of-database

Druhým způsobem je uložení samotných rastrových dat mimo databázi v souborovém systému, kdy databáze má uložené pouze indexovaný geografický rozsah korespondující s daným rastrem, informace o cestě k samotnému rastru v souborovém systému a informace o rastru samotném.

Výhody:

- rastry je možné sdílet s většinou aplikací – ty nepotřebují znát postup, jak číst data přímo z databáze
- databáze je mnohem menší než v případě uložení rastrů uvnitř databáze

Nevýhody:

- tento model uložení není pořádně otestován
 - veškeré výhody samotného uložení v databázi se ztrácí
 - rastry uložené v souborovém systému nemají vazbu s databází – pokud je vymazán rastr ze souborového systému, záznam v databázi v tu chvíli ztrácí smysl – vazba s cestou k souboru je neplatná
 - databáze musí mít přístup k rastrům uloženým v soubor. systému a mít příslušná práva
 - analýzy rastrů zatím nejsou k dispozici a až budou vytvořeny, budou mnohem pomalejší než při uložení uvnitř databáze.
 - model uložení *out-of-database* zatím nepodporuje vytváření náhledů, které jsou pro rychlost zobrazování v mapových službách podstatné
- (Obe; Hsu, 2011)

3.1.3. Obecné zhodnocení na úrovni uložení rastrů v PostgreSQL

Výhody:

- rastry jsou na jednom místě s ostatními daty, vektory a atributy – jednoduché analýzy vektor/rastr/atributy
- základní operace s daty jsou prováděny v databázi – není potřeba další prostředí – exporty rastrů, prohlížení, modifikace, atd.
- multi-uživatelské prostředí
- jednoduchý vývoj klientských aplikací – operace s daty jsou součástí databáze – klient může provádět operace, které už někdo napsal, geometrické operace jsou, co se vývoje kódu týká, velmi náročné
- bezpečnost

(Blasby, 2001; Obe, Hsu, 2011; Racine, 2011)

Nevýhody:

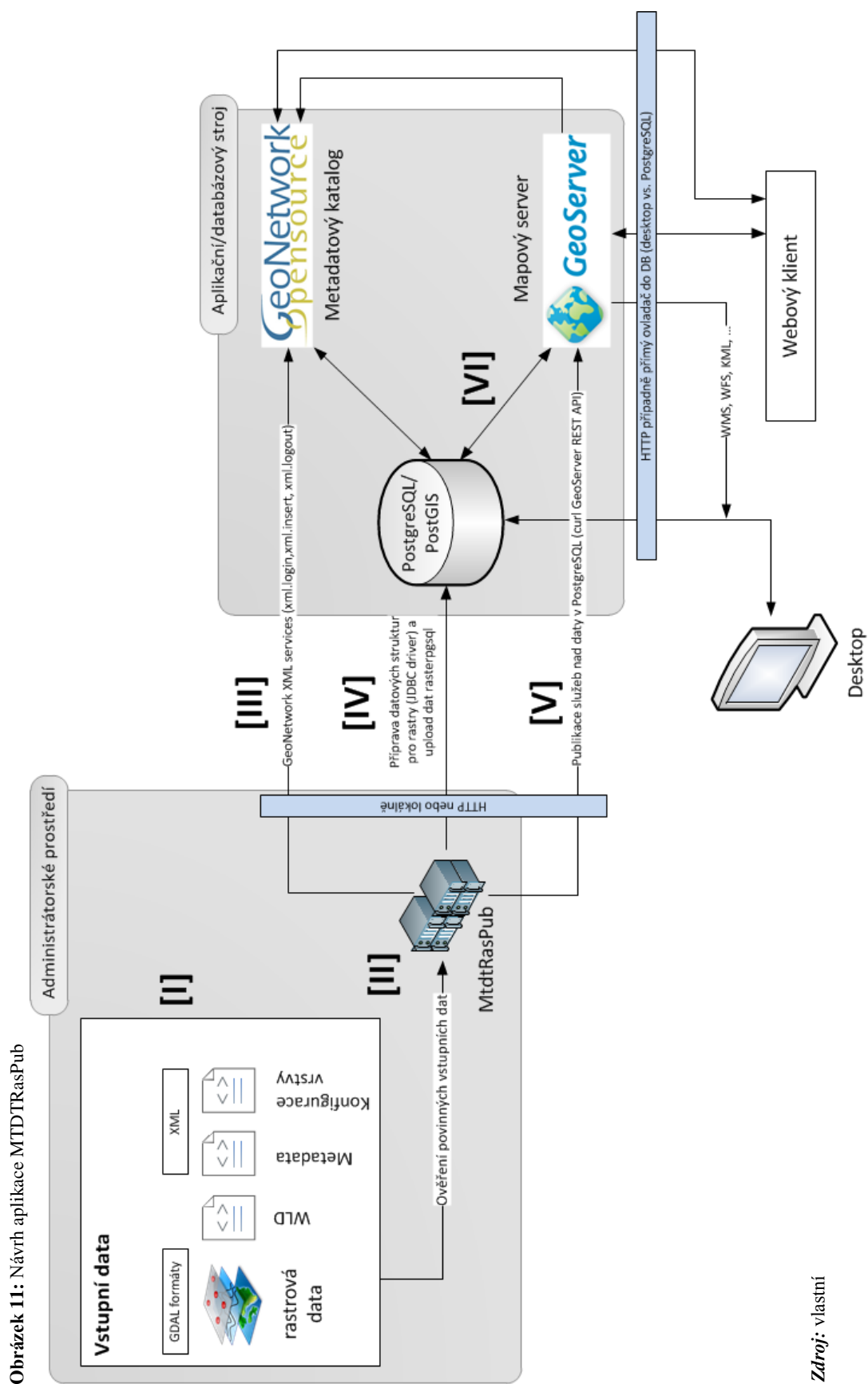
- implementace může být náročnější
- obecně je uložení v databázi vyžadující speciální struktury pomalejší než uložení přímo v souborovém systému – výrazně by se mohlo vylepšit s příchodem PostgreSQL 9.2, které bude podporovat prostorový index SP-GiST (Spatial Partitioning Generalized Search Tree), který podle prvních testů až trojnásobně zrychluje vytváření indexu a až šestnásobně dotazování na indexovaná data (Bartunov; Sigaev, 2011)
- nekompatibilita s některými GIS aplikacemi – ale vzhledem k času, po který se PostGIS raster vyvíjí, se nechá očekávat zlepšení podpory
- nezkušenost uživatelů s něčím ne úplně standardním

(Blasby, 2001)

Je nutné se tedy rozhodnout, jak se k celému konceptu postavit, zda se chceme zabývat databází a postavit se tak před úkol, který bude do budoucna využitelný mnoha směry, jak uložení, tak analýzy dat, tak jako část SDI nebo se pouze spokojit s konstatováním, že je souborové uložení rychlejší a dál se tímto tématem nezabývat. Ale proto by toto téma nevzniklo a přínosy a výhody, které jsou v této kapitole zmíněny jsou dostatečným zdůvodněním, proč se tímto tématem zabývat a pokusit se vyřešit otázky, které práce nastolila.

3.2. Návrh řešení a definice dílčích problémů

V následující kapitole je popsán kompletní návrh řešení a definice dílčích problémů, které jsou nutné pro zdárné vyřešení celé aplikace. Obrázek 11 detailně popisuje návrh celého řešení, nad kterým bude provedena implementace, včetně popisu jednotlivých částí, které se aplikace budou účastnit. Obrázek je orientovaný na šířku se čtením zleva k pravé straně, kde se nachází výstupy směrem do webového prostředí a desktop.



Na obrázku 11 je definováno šest funkčních celků, označených římskými číslicemi, u kterých bylo třeba řešit dílčí úkoly.

Vstupní data [I]

Bylo třeba definovat formáty a možnosti formátů na vstupu. Obecně lze vkládat jakákoliv rastrová data, která jsou podporována GDAL knihovnou. Dalšími daty na vstupu jsou referenční soubor *.wld, sloužící ke georeferencování rastru na vstupu do databáze, metadata ve formátu XML a konfigurace samotné vrstvy pro mapový server (souřadnicový systém, název, jméno datového prostoru mapového server - *Workspace*), také ve formátu (m)XML. M(XML) je pouze modifikace XML změnou jeho formátu kvůli aplikačnímu chování.

Aplikace MTDTRasPub – vlastní aplikace [II]

Aplikace zastřešující kompletní řešení pro publikaci rastrových dat z PostgreSQL pro mapový server GeoServer a metadatový katalog GeoNetwork opensource. Na této úrovni se také řeší prolínání metadatového záznamu a informací získaných z rastrových podkladů jako je například prostorový rozsah dat, vy publikované služby zobrazující data a další.

Publikace metadat [III]

GeoNetwork disponuje více možnostmi jak s katalogem komunikovat, bylo nutné tyto možnosti prostudovat a pro jednu z nich se rozhodnout. Zde nebyla dodržena specifikace podle standardů OGC, kdy bylo vycházeno ze staršího zdroje (Ožana, 2007). V tomto zdroji je zmiňována implementace nepovinných CSW operací v GeoNetworku pro editaci metadat jako neimplementovaná. Tato informace nebyla ověřena a byly použity namísto služby CSW GeoNetwork XML services, konkrétně operace *metadata.insert*. Na straně druhé, toto nedodržení standardu přineslo pro aplikaci některé výhody, protože přes XML services lze ovládat celý katalogový server. Také je zde potřeba definovat novou databázi a to sice PostgreSQL na rozdíl od databáze McKoi, která je standardní pro GeoNetwork (Doeleman; García, 2010). Na vstupu jsou očekávány některé ze standardů podporovaných GeoNetwork opensource.

Vytvoření datových struktur a import dat do PostgreSQL [IV]

V této části práce byla využita část knihovny GeoTools Image Mosaic JDBC plugin (dále jen IMJDBC), kterou využívá GeoServer pro komunikaci s rastry uloženými v PostGIS. Cílem bylo navrhnout takovou strukturu, která by umožňovala co nejvíce typů výstupu z databáze, aby data mohla být zobrazována jak ve webových službách, tak desktopových aplikacích. Bylo přepracováno generování rastru na vstupu do databáze. Původní postup podle (GeoServer, 2012 [a]) byl změněn a použity byly jen původní databázové struktury, které byly rozšířeny. Postup importu dat do databáze byl přepracován pro standardní importovací nástroj *raster2pgsql* a modifikace databáze byla provedena za použití funkcí PostGIS raster.

Publikace mapové služby [V]

Zřejmě nejobtížnější úkol v kombinaci s IMJDBC pluginem pro GeoServer. Není znám žádný postup, jak automaticky vytvořit *DataStore* (datový zdroj Geoserveru připojený k rastrovým

podkladům v PostgreSQL). Existují manuální postupy, ale nikoliv automatické (GeoServer, 2012 [a]). Zde se plně ukázala výhoda open source modelu a prostudováním zdrojového kódu nalezena cesta, jak *DataStore* automaticky vytvořit. Pro publikaci metadat je používáno rozhraní REST API ovládané přes nástroj CURL z kódu Javy. Dalším problémem bylo nezpřístupnění parametrů průhlednosti pozadí mapové vrstvy, kdy vypublikovaná vrstva měla kolem sebe bílé okolí, kterým překryla všechny ostatní vrstvy, které byly v prostředí klienta připojeny (více viz kapitola 5).

Komunikace GeoServeru s PostGIS raster [VI]

Obecně nejsou známy žádné testy ani vyhodnocení rychlostí, které by prokazovaly výhody či nevýhody použití tohoto modelu. Proto bude v kapitole 5 zhodnoceno, jestli je tento přístup vhodný pro nasazení, které je v práci popisováno. Dále budou provedeny základní testy rychlosti na různých datových sadách. Tyto testy by měly potvrdit nebo vyvrátit, zda rychlost bude skutečně limitujícím parametrem a nevyplatí se navzdory dalším výhodám zmiňovaným na začátku kapitoly tento model využívat.

Uživatelské kompozice

Problematika uživatelských kompozic není v obrázku 11 uváděna, protože tato funkční část by zasahovala do více částí systému. Bylo by nutné ji implementovat na více úrovních. První úroveň na straně klienta, kde by uživatel na základě aktuálního stavu klienta (načtené vrstvy, aktuální měřítko, souřadnicový systém, atd.) musel být schopen uložit aktuální stav v podobě WMC dokumentu. Druhou úrovní by byla databáze, kde by byl dokument uložen. Protože tato část implementace by byla značně náročná a závislá na uživatelských účtech, které tato práce neřeší, je tento krok nahrazen pomocí skupin vrstev, které poskytuje GeoServer a umožňuje jejich publikaci pomocí REST API.

Více ke zmiňovaným částem aplikace v kapitole 5.

KAPITOLA 4

Implementační prostředí

Vzhledem k obrovskému rozvoji open-source technologií na poli GIS v posledních letech a značnému rozsahu technologií, které je potřeba pro úspěšný provoz vyvíjené aplikace (dále jen MTDT RasPub), považuji tuto kapitolu jako jednu z klíčových k pochopení a úspěšnému fungování celé aplikace.

V podstatě denně jsou vydávány nové verze software, který je v této práci využíván, proto nebylo možné stále se držet poslední vydané verze, protože některé aktualizace přinášely problémy, kdy nebylo možné udržet stále funkční kód vlastní aplikace napsaný nad aktualizovaným prostředím. Z tohoto důvodu jsem se rozhodl „zmrazit“ prostředí pro implementaci mé diplomové práce ve stavu, který popisuje aktuální kapitola a dále prostředí v rámci diplomové práce neaktualizovat.

Samozřejmostí je možnost aktualizace samotného kódu do budoucna, společně s aktualizacemi využívaného softwaru.

Nedílnou součástí této kapitoly je detailní popis implementačního prostředí, jeho instalace, specifika nastavení a problémy, se kterými jsem se během přípravy prostředí setkal.

Za zmínku stojí fakt, že za tak krátké období, kdy tato práce vznikala (listopad 2011 – červenec 2012), došlo k mnohým zásadním změnám, které výrazně usnadnily instalaci a nastavení některých komponent, které práce využívá a zároveň také instalaci na 64 bitové prostředí.

Implementace a testování aplikace budou prováděny v prostředí MS Windows 7 Professional, protože většina serverů na KAGIK běží právě na platformě Windows, kde se také předpokládá pozdější nasazení. Vzhledem k tomu, že jsou používány výhradně volně dostupné technologie nezávislé na platformě, je možné použít program po malých úpravách i na jiných platformách.

4.1. Databázové prostředí (PostgreSQL s prostorovou nadstavbou PostGIS)

PostgreSQL je základním stavebním kamenem této diplomové práce.

4.1.1. Instalace PostgreSQL 9.1 +

Instalace PostgreSQL na platformu Microsoft Windows doznala velkého pokroku a zjednodušení poté, co PostgreSQL začalo podporovat nativně Windows platformu, což znamenalo od verze PostgreSQL 8.0 a vyšší. Před verzí 8.0 bylo nutné využívat nástrojů Cygwin pro běh PostgreSQL na platformě Windows (PostgreSQL, 2005).

Dalším velkým skokem k uživatelskému komfortu bylo vydání instalačního balíku PostgreSQL firmou EnterpriseDB, který je dostupný od verze PostgreSQL 8.3.x a přináší instalaci, která je obvyklá pro platformu Windows formou instalátoru.

Problémem byl také fakt, že PostgreSQL až do poslední verze nepodporovala běh databáze pod 64 bitovými systémy jako 64 bitová aplikace. Databáze byla schopna běžet pod 64 bitovými systémy, ale pouze jako 32 bitová aplikace (EnterpriseDB, c2011), což by se mohlo nepříznivě projevit při zpracování náročnějších operací s rastry do budoucna.

Po instalaci bylo nutné ještě udělat následující operaci jako administrátor v příkazovém řádku.

```
setX PATH "%PATH%;C:\Program Files\PostgreSQL\9.1\bin" /m
```

Tato operace nastaví proměnnou prostředí PATH tak, že zachová všechny původní (%PATH%) a přidá jednu novou (C:\Program Files\PostgreSQL\9.1\bin). Tato proměnná je přidávána z důvodu dostupnosti příkazu PSQL pro ovládání databáze PostgreSQL přes příkazovou řádku.

Jednoduchým ověřením, že příkaz nastavení systémové proměnné proběhl správně, je spuštění příkazové řádky MS Windows (cmd.exe) a vyvolání příkazu *psql --help*, který by měl vypsát manuálovou stránku příkazu *psql*.

4.1.2. Instalace PostGIS 2.0.0 s podporou PostGIS raster

Instalace PostGIS procházela podobným vývojem jako instalace databáze PostgreSQL. První verze byly uživatelsky náročné na instalaci, hlavně v prostředí Microsoft Windows, pro které nebyl PostGIS nativně podporován. PostGIS je v posledních verzích možné nainstalovat pomocí tzv. aplikace „Stack Builder“, která je součástí předchozí instalace PostgreSQL. „Stack Builder“ je instalátor doplňků třetích stran do prostředí databáze PostgreSQL.

Dalším problémem do nedávné minulosti způsobovala nekompatibilita 64 bitové PostgreSQL a 32 bitové nadstavby PostGIS, kdy databáze samotná již byla k dispozici v 64 bitové verzi, ale PostGIS nikoliv. Proto bylo nutné vyšší verzi PostgreSQL i na 64 bitových strojích snížit na 32 bitovou, aby bylo možné následně nainstalovat 32 bitovou verzi PostGIS. Verze PostGIS pro 64 bitové stroje byla vydána až v dubnu 2012 (Pgfoundry, 2007).

Ve verzi 2.0.0 byl zpřístupněn PostGIS raster jako součást PostGISu, kdežto v předešlých verzích bylo nutné instalovat PostGIS raster složitější cestou jako nadstavbu nad již běžící databází PostgreSQL (Obe; Hsu, 2011).

Během instalace je automaticky nabídnuta příprava databáze podporující nadstavby PostGISu. Pokud je vybrána možnost vytvoření databáze, usnadní se vytváření dalších databází s podporou PostGIS, protože lze původně vytvořenou databázi využít jako šablonu pro databáze následující. Další postup popisuje, jak vytvořit databázi s podporou PostGISu bez nutnosti používat grafické prostředí pro správu databáze (např. PgAdmin III).

4.1.2.1. Konfigurace databáze s podporou PostGIS

Velmi jednoduše lze vytvořit databázi pomocí šablony z databáze vytvořené při instalaci PostGISu (standardně pojmenovaná *postgis20*). Pokud je pro vytvoření databáze používána šablona jiné databáze, musí být uzavřena všechna připojení k databázi, která je používána jako šablona. Tento postup je jednoduchý, má však svá úskalí, která jsou popsána na následujících řádcích, které by měly usnadnit případné hledání chyb, které jsou nezřídka velmi špatně zdokumentované a popsané.

```
select * from pg_stat_activity; -- Vypsání aktivních připojení v DB

select pg_terminate_backend(1576) from pg_stat_activity; -- Zastavení
jednoho procesu podle procpid

SELECT procpid, (SELECT pg_terminate_backend(procpid)) as killed from
pg_stat_activity WHERE datname LIKE 'postgres'; --Zastavení procesů podle
podmínky where
```

Vytvoření databáze podle šablony.

```
CREATE DATABASE "SamplePostGISDbConf"
WITH ENCODING='UTF8'
OWNER=postgres
TEMPLATE=postgis20
CONNECTION LIMIT=-1
TABLESPACE="Test";
```

Po vytvoření této databáze je možné zkontrolovat, zda vytvoření proběhlo v pořádku a PostGIS je připraven k práci. Následující dotaz do databáze, ve které má být existence PostGIS ověřena, by měl vypsát informace v tabulce 5.

```
psql -d SamplePostGISDbConf -U postgres -c "select
postgis_full_version(), postgis_raster_lib_build_date(),
postgis_raster_lib_version();"
```

Tabulka 5: Výsledek dotazu pro ověření instalace PostGIS

postgis_full_version	postgis_raster_lib_build_date	postgis_raster_lib_version
POSTGIS="2.0.0 r9605" GEOS="3.3.3-CAPI-1.7.4" PROJ="Rel. 4.8.0, 6 March 2012" GDAL="GDAL 1.9.0, released 2011/12/29" LIBXML="2.7.8" LIBJSON="UNKNOWN" RASTER	2012-04-04 19:24:50	2.0.0 r9605

Zdroj: vlastní

V případě, že zdrojová databáze s podporou PostGIS není dostupná, je nutné postupovat následovně.

1. Vytvoření prázdné databáze bez podpory PostGIS.

```
psql -d postgres -U postgres -c "CREATE DATABASE SamplePostGISDbConf_man
WITH OWNER=postgres TABLESPACE='\"Test\"';"
```

2. Vytvoření podpory PostGIS .

```
REM Nahraje objekty a funkce PostGIS do databáze
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\postgis.sql"

REM Nahraje definici EPSG kódů souřadnicových systémů
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\spatial_ref_sys.sql"

REM Vytvoří komentáře k funkcím a objektům PostGIS
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\postgis_comments.sql"

REM Nahraje objekty a funkce PostGIS Raster do databáze
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\rtpostgis.sql"

REM Vytvoří komentáře k funkcím a objektům PostGIS Raster
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\raster_comments.sql"

REM Vytvoří podporu topologie v databázi
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-2.0\topology\topology.sql"

REM Vytvoří komentáře k funkcím topologie
psql -d samplepostgisdbconf_man -U postgres -f "C:\Program
Files\PostgreSQL\9.1\share\contrib\postgis-
2.0\topology\topology_comments.sql"
```

Tento postup bylo nutné používat ve verzích PostgreSQL 9.1 a nižších. Od verze 9.1 lze použít pro vytvoření podpory PostGIS v databázi tzv. *Extension* a to následovně.

1. Vytvoření prázdné databáze bez podpory PostGIS.

```
psql -d postgres -U postgres -c "CREATE DATABASE
SamplePostGISDbConf_extensions WITH OWNER=postgres TABLESPACE='\"Test\"';"
```

2. Vytvoření podpory PostGIS pomocí CREATE EXTENSION .

```
REM Nahraje objekty a funkce PostGIS do databáze
psql -d samplepostgisdbconf_extensions -U postgres -c "CREATE EXTENSION
postgis;"

REM Nahraje objekty a funkce PostGIS topology, který je odděleným balíkem
PostGIS do databáze
psql -d samplepostgisdbconf_extensions -U postgres -c "CREATE EXTENSION
postgis_topology;"
```

Takto připravené databázové prostředí je plně postačující pro další práci, kde budou funkce a části PostGIS 2.0.0 včetně PostGIS Raster využívány.

Je velmi nevhodné nechat existující databáze živé a běžící v produkčním prostředí pod tzv. super uživatelem „postgres“ z bezpečnostního hlediska. Proto bude pro další práci vytvořena databáze, kde vlastníkem bude uživatel s menším oprávněním.

4.2. Mapový server (GeoServer 2.1.4 +)

Mapový server je jedním z dalších stavebních prvků tohoto modelu. Slouží k publikaci mapových podkladů v prostředí internetu a byl jedním z velkých výzkumných úkolů této práce vzhledem k neprobádanosti komunikace mezi mapovým serverem GeoServer a databázovou platformou PostgreSQL v kontextu k obrazovým datům.

Vzhledem k účelům vývoje aplikace nebyl nasazován ani GeoServer ani v následující kapitole zmiňovaný GeoNetwork opensource v servletovém kontejneru robustního webového serveru, jakým je například Apache Tomcat, ale obě instance, obou serverů běží pod vlastním webovým serverem, kterým je Jetty server.

4.2.1. Instalace GeoServeru

Instalace GeoServeru je velmi jednoduchá, je třeba ovšem doplnit GeoServer o funkcionalitu podporující práci s rastry uloženými v databázi.

GeoServer, ačkoliv běží v prostředí 64 bitových MS Windows, vyžaduje pro běh JRE (Java Runtime Environment) v 32 bitové verzi. Tato prerekvizita vzniká společně s požadavkem, aby GeoServer běžel jako služba pod operačním systémem.

Při instalaci mapového serveru je již dobré dopředu uvažovat o tom, na jakém portu bude provozován vzhledem ke vztahu k ostatním aplikacím běžícím na stejném hardwaru. Standardně je nabízena instalace pro běh na portu 8080, což je ale vhodné pouze pokud GeoServer běží ve webovém kontejneru Apache Tomcat, pokud běží se samostatným webovým serverem, je vhodné použít jiný port např. 8081 a 8080 ponechat pro běh právě Apache Tomcat nebo GeoNetworku.

4.2.2. Instalace podpory PostGIS raster do GeoServeru

Podpora rastrových dat uložených v PostGIS není standardní součástí instalace, proto je nutné ji doplnit. Pro instalaci bylo nutné použít verzi 2.1.4. a vyšší, protože do nižších verzí se mi nepodařilo přidat doplněk podporující práci s PostGIS raster.

Jak bylo zmíněno v předešlé kapitole, podpora pro rastry uložené v PostGIS je zprostředkovávána pomocí pluginu IMJDBC, kterou je možné použít pro různé typy databází (Oracle, DB2, MySQL a další, které podporují datový typ BLOB).

Instalace u této komponenty spočívá pouze v nakopírování knihovny do adresáře *WEB-INF* v instalačním adresáři GeoServeru a následný restart jak samotného mapového serveru, tak i

služby běžící pod Windows, poté se doplněk objeví v možnostech založení nového *DataStore* v administračním prostředí GeoServeru.

4.3. Katalogový server/klient GeoNetwork

Instalace GeoNetwork opensource v prostředí MS Windows je velmi obtížná, proto by následující řádky měly usnadnit tuto operaci.

4.3.1. Instalace GeoNetworku s podporou PostgreSQL

Po instalaci samotného GeoNetworku nelze spustit hlavní okno aplikace standardně dostupné na adrese `http://localhost:8080/geonetwork`. Pokud je aplikace správně spuštěna hned při spuštění se objeví okno s logovacím dialogem aplikace, tento log je výrazným vodítkem při zprovoznování funkční podoby GeoNetworku.

V prvním kroku je třeba zkontrolovat umožnění přístupu do databáze z prostředí localhost, toto je možné zkontrolovat v soubor `C:\Program Files\PostgreSQL\9.1\data\pg_hba.conf`, který by měl vypadat následovně.

```
local    all all trust
# IPv4 local connections:
host     all all 127.0.0.1/32 md5
```

Dalším krokem je vytvoření databáze pro struktury GeoNetworku a vytvoření samotných struktur, včetně naplnění testovacími daty, pokud nejsou správně naplněné systémové proměnné z kapitoly 4.1, tak následující operace nepůjdou v příkazovém řádku spustit.

```
postgres=# CREATE USER geonetwork WITH PASSWORD 'geonetwork';
postgres=# CREATE DATABASE geonetwork WITH OWNER = geonetwork ENCODING
'UTF8';
psql -d geonetwork -U geonetwork -W -f C:\Program
Files\geonetwork\web\geonetwork\WEB-INF\classes\setup\sql\create\create-
db-postgres.sql
psql -d geonetwork -U geonetwork -W -f C:\Program
Files\geonetwork\web\geonetwork\WEB-INF\classes\setup\sql\data\data-db-
postgres.sql
```

Poté je nutné ještě nastavit připojení GeoNetworku k PostgreSQL v konfiguračním souboru umístěném v `C:\Program Files\geonetwork\web\geonetwork\WEB-INF\config.xml`.

(dosavadní postup podle Doeleman; García, 2010)

V prostředí Windows 7 je třeba se vyvarovat následujícím chybám, které způsobují nefunkčnost GeoNetworku. Veškerá následující chybová hlášení se objeví v logu GeoNetworku, který je vypisován na standardní výstup při spuštění serveru.

- Protože je GeoNetwork připraven pro práci v servletovém kontejneru, je připraven v aplikaci i archiv s GeoServerem, ale vzhledem k tomu, že naše instance běží na webovém Jetty serveru, je tento archiv nepoužitelný a GeoNetwork vrací chybu.

Protože je k dispozici jiná instance GeoServeru, je možné tuto vymazat `C:\Program Files\geonetwork\web\geoserver.war`.

- Jetty server zapisuje logy do `C:\Program Files\geonetwork\jetty\logs`, ale defaultně nemají uživatelé do této složky povolen zápis, proto je nutné povolit zápis do této složky uživatelskému účtu, pod kterým běží GeoNetwork.
- Stejná práva pro zápis jsou třeba přidat do následující složky `C:\Program Files\geonetwork\web\geonetwork\WEB-INF\classes`.
- Poté se GeoNetwork snaží vytvářet indexy v následujících dvou složkách, ty je nutné vytvořit a přiřadit příslušná práva `C:\Program Files\geonetwork\web\geonetwork\WEB-INF\lucene\spatial` a druhou složkou je `C:\Program Files\geonetwork\web\geonetwork\WEB-INF\lucene\nonspatial`.

Po provedení těchto úprav by měl log při startu GeoNetwork vypsat pouze informace, ale neměla by se v něm objevit chybová hlášení. Tento postup byl ověřen na 2 instancích Windows 7 Professional. Nefunkčnost GeoNetwork se projevovala tím, že nebylo možné se přihlásit jako základní uživatel *admin/admin*.

Pokud předešlé kroky byly v pořádku dokončeny, posledním problémem je neexistence mapových zdrojů v prostředí GeoNetwork. Protože byl odstraněn původní GeoServer, který je připraven pro běh pod webovým serverem Apache Tomcat, který není v této práci použit, je logické, že mapové podklady se nemají do prostředí GeoNetwork odkud nahrávat. Protože byla připravena nová instance GeoServeru z kapitoly 4.2.1, stačí převzít mapové podklady z instalačního adresáře GeoNetwork a vy publikovat nové vrstvy v běžící instanci GeoServeru. Poté je třeba ještě přepsat konfiguraci připojení základních vrstev do mapového klienta GeoNetwork v `C:\Program Files\geonetwork\web\geonetwork\WEB-INF\config-gui.xml`.

Úplně nakonec, pokud byly úspěšně vyřešeny všechny předchozí problémy, je vhodné vypnout vypisování logu na standardní výstup v `C:\Program Files\geonetwork\bin\start-geonetwork.bat`.

4.4. Desktopový klient a závislosti

V této kapitole lze problémy obejít jednoduchým způsobem a to sice tak, že použijeme OSGeo4W instalátor aplikací pro Windows (OSGeo4W, 2012), kde pro naše účely postačuje vybrat možnost „Express desktop install“ a na další kartě dvě položky QGIS a GDAL. Tímto je ošetřena většina závislostí, které budeme potřebovat pro běh vyvíjené aplikace MTDTRasPub.

V tomto kroku je třeba ověřit, zda máme dostupné potřebné funkce a knihovny.

raster2pgsql

Raster2psql je importovací nástroj rastrových formátů do PostgreSQL. Stačí ověřit, zda v příkazovém řádku proběhne v pořádku příkaz `raster2pgsql -G`, který vypíše seznam podporovaných formátů pro import do PostGIS. Tento přehled je velmi rozsáhlý a obsahuje kolem stovky formátů (viz příloha 2).

CURL

Je program pro práci s http protokolem z příkazové řádky (Curl, 2012). V aplikaci je používán pro komunikaci s REST API GeoServeru skrze Javu.

Pokud jsou splněny všechny požadavky z kapitoly 4, aplikace MTDTRasPub by měla mít vše nutné ke svému běhu.

KAPITOLA 5

Vývoj programu MTDTRasPub

Aplikace byla vyvíjena s přihlédnutím na celkový koncept navrhované SDI na KAGIK UK podle (Grill ... [et. al], 2010), kdy jsou použity velmi podobné technologie a jádrem je také PostGIS a metadatový katalog GeoNetwork. Dalším prvkem, který do vytváření aplikace vstoupil, byl projekt oživení mapové sbírky na Přírodovědecké fakultě UK (Novotná, 2011), který by podobnou technologii také využil.

5.1. Ověření manuální cestou

V první fázi bylo potřeba ověřit, zda je možné vypublikovat mapové služby manuálně a jak k tomuto přistoupit. Záměrně zde neuvádím publikaci metadat, protože u metadat je cesta jak vypublikovat data známá a bylo ji nutné pouze automatizovat.

Problém tedy nastával hlavně na úrovni mapového serveru společně s rastry uloženými v databázi. Protože byl jako mapový server zvolen GeoServer, který podporuje jako datový zdroj rastry uložené v databázi, bylo nutné prozkoumat na vývojářské úrovni IMJDBC plugin pro GeoServer (Mueller, 2009), který zprostředkovává komunikaci mezi PostgreSQL a GeoServerem. Tak bylo možné pochopit jeho fungování, aby bylo možné z těchto vědomostí vyjít a vytvořit automatickou cestu importu samotných dat do databáze, přípravu databázových struktur a zároveň vypublikovat i služby postavené na importovaných datech v prostředí GeoServeru.

Ověření manuální cestou se podařilo na základě manuálu, který poskytuje dokumentace k samotnému GeoServeru (GeoServer, 2012). Ovšem tato cesta nebyla použitelná vzhledem k nutnosti automatické publikace a dále protože vyžadovala administrátorské přístupy do systému. Druhou obrovskou nevýhodu znamenalo to, že rastrová data jsou nahrávána rovnou jako binární data, tudíž s nimi nelze udělat v databázi žádné další operace a slouží pouze k publikaci webových služeb. Tento způsob byl nevhodný a nebylo možné ho dále řídit a s daty pracovat. Posloužil jako odrazový můstek a základ pro vlastní rozvoj aplikace.

5.2. Popis funkcionality a testování

Jedná se o program napsaný v Javě, který zprostředkovává publikaci metadat a dat do databázového programu PostgreSQL a dále do katalogového a mapového serveru, jak zobrazuje obrázek 11. Jedná se o první verzi, která vznikla jako výsledek této diplomové práce.

Program má vlastní konfiguraci ve formátu XML. Tato konfigurace musí být umístěna ve stejné složce jako samotný program a musí se jmenovat *AppConfig.xml*. Definuje základní parametry programu, které jsou nutné k běhu (viz příloha 3). Tento princip je velmi snadno rozšiřitelný o potřebné položky v případě rozvoje aplikace bez velkých zásahů do kódu aplikace.

5.2.1. Vstupní data a podmínky

V první fázi programu je třeba definovat vstupní data. Program na vstupu očekává 4 soubory. Samotný rastr, referenční soubor (tzv. worldfile) s příponou **.wld*, který georeferencuje daný rastr a zároveň ho požaduje importní nástroj rastrů do PostGIS *raster2pgsql*, pokud georeference neexistuje v hlavičce rastru. Dalším souborem jsou metadata ve formátu XML, a poslední položkou jsou potřebné položky pro mapový server **.mxml* (příloha 4). Složka, kde jsou rastry pro import vyhledávány je definována v *AppConfig.xml* (příloha 3). Podmínkou je, že se všechny soubory patřící k jednomu rastru musí jmenovat stejně, například ortofoto.png, ortofoto.xml, ortofoto.mxml a ortofoto.wld. Pokud toto nebude splněno, aplikace vrátí hlášení o skutečnosti, že nemá k dispozici validní rastry k importu. Aplikace zatím podporuje názvy pouze malými písmeny, bez diakritiky a mezer. Jména rastrů jsou používána v databázi a to by pak mohlo způsobovat problémy. Ve složce s rastry je možné vytvořit složku s názvem „imported“, kam jsou odsouvány již importované rastry. Pokud složka neexistuje, rastry i ostatní soubory zůstávají po importu v původní složce. V tomto případě rastry budou naimportovány znovu při příštím spuštění aplikace. Kromě faktu, že toto zvýší čas zpracování, se nic zásadního nestane. Aplikace je napsána tak, aby přepsala data, která mají stejný název, a v případě metadat stejný identifikátor (UUID). Jako vstupní rastrové formáty jsou zatím podporovány JPEG, PNG a TIFF.

Pokud jsou tyto podmínky splněny a zároveň je v pořádku prostředí, které je definováno v kapitole 4, je možné spustit aplikaci pro import dat. V prvním kroku dojde k ověření dat definovaných v předchozím odstavci. Pokud se najdou data vhodná pro import, je spuštěn. Data jsou načtena do prostředí Javy a zde začíná práce se samotnými daty.

Dochází k prolínání dat z rastrových podkladů, které jsou získávána nástroji GDAL a následně kombinována s metadaty, která jsou posléze vypublikována.

5.2.2. Metadata

Vzhledem ke komplikovanosti metadatových záznamů, které mnohdy obsahují stovky a více řádků XML, jsem se rozhodl tuto pasáž řešit zatím přímo z kódu aplikace a nikoliv načtením externích metadat. Přestože jsou metadata na začátku aplikace validována ve smyslu existence, v této verzi aplikace nejsou používána.

Důvodem, proč je takto aplikace vyřešena, je složitost metadatových XML, kdy se stává, že jsou metadatová XML nevalidní a v aplikaci dochází k chybám.

Validní XML je proto uloženo přímo v kódu aplikace a jsou v něm dynamicky měněny informace, které jsou získány z rastrů a následně aplikovány při publikaci do metadatového katalogu.

V tomto kroku by aplikace měla komunikovat s katalogovým serverem pomocí CSW služby. Vzhledem ke složitosti implementace této služby jsem tuto funkci nahradil pomocí jiné služby metadatového katalogu a to sice jednou z možností GeoNetwork XML services *metadata.insert* (GeoNetwork opensource, c2010). Tato služba disponuje vlastností, která pokud najde metadata se stejným identifikátorem, je schopná je nahradit a ne vytvořit nový metadatový záznam. To velmi zjednodušuje implementaci, protože nemusí být hlídáno, jestli existují data, a co se s nimi musí provést, což by vyžadovala implementace CSW.

Vzhledem k faktu, že v oblasti metadat není moc dalších možností mimo GeoNetwork, které by šly využít v konceptu této aplikace, není implementace CSW pro aktuální využití nutná. Pro obecné použití by byla vhodnější.

Je třeba počítat s tím, že jakékoliv operace s metadaty jsou operace pod přihlášeným uživatelem, proto by bylo nutné část XML services použít i s CSW a to operace *user.login*. Zde vstupem je uživatelské jméno a heslo a výstupem SID (Session ID), které je používáno při operaci *metadata.insert* a nakonec operace *user.logout*, která odhlásí aktuální SID ze systému.

Tento postup má jednu nevýhodu. Metadata, která jsou vypublikována, jsou dostupná pouze uživateli, který je vytvořil. Pro zpřístupnění metadat všem uživatelům by bylo nutné po vytvoření metadat ještě použít jednu z balíků služeb *metadata ownership services*, která umožňuje manipulaci s vlastnictvím metadat a delegaci práv.

5.2.3. Datové struktury pro rastry

Zde bylo nutné vymyslet vlastní přístup, protože importní nástroj, který je součástí IMJDBC pluginu pracuje odlišným způsobem, než potřebujeme k využití pro automatickou publikaci a vyžaduje administrátorská oprávnění pro práci.

Protože tento plugin očekává v databázi strukturu, kdy první tabulka (tabulka 6) zobrazuje tzv. mateřskou tabulku, která sdružuje vazby na dceřinné tabulky (tabulka 7), ve kterých jsou uloženy dlaždice z jednotlivých hladin pyramid a kde již jsou uložena samotná data. Rastrová data jsou ale uložena jen v binárním formátu (sloupec *data[bytea]*), kdy právě IMJDBC při importu dat použije funkci *gdal_retile* a vytvoří si dlaždice již v souborovém systému, které pak importuje, ale pouze jako binární formát, ze kterého už na úrovni databáze nelze vytvořit vnitřní formát PostGIS raster pro rastry. Detailní popis struktur databáze ukazuje (Mueller, 2009).

Tabulka 6: Struktura tabulek v PostGIS po publikaci IMJDBC – mateřská tabulka

name [PK] character(64)	spatialtable [PK] character varying(128)	tiletable [PK] character varying(128)
layer_marklissa	marklissa	marklissa
layer_marklissa	o_2_marklissa	o_2_marklissa
layer_marklissa	o_4_marklissa	o_4_marklissa

*Zdroj: vlastní***Tabulka 7:** Struktura tabulek v PostGIS - tabulka o_2_marklissa

filename text	data bytea	geom geometry(MultiPolygon,4326)
marklissa.png	<binární data>	0106000020E610000001000000
marklissa.png	<binární data>	0106000020E610000001000000
marklissa.png	<binární data>	0106000020E610000001000000

Zdroj: vlastní

Tento způsob publikace je ale nevýhodný, protože data nejsou dále použitelná pro žádný jiný účel než společně s IMJDBC. Proto byl navržen vlastní postup pro plnění databáze, který zajišťuje vytvoření struktur, které požaduje IMJDBC a zároveň vytvoření vnitřního formátu rastru pro PostGIS.

Pro tento účel byl použit standardní nástroj pro import dat do databáze *raster2pgsql*, kdy rastr je naimportován následujícím způsobem a zároveň je upravena struktura databáze. *Raster2pgsql* disponuje také možností pro vytváření dlaždic rastrů a pyramid, proto bylo vhodné použít tento nástroj a pomocí funkcí PostGIS raster pak sestavit binární data. Rastr je v tomto případě importován do databáze a následně z něj pomocí funkce *ST_asPNG* nebo *ST_asTIFF* je vytvořena binární podoba a doplněn sloupec do tabulky. Protože IMJDBC používá k usazování dlaždic geometrie reprezentující rastr, bylo nutné vytvořit také polygon, který reprezentuje vždy jednu původní dlaždici rastru pomocí funkce *ST_Envelope* z interního formátu PostGIS raster a vytvořit sloupec s touto geometrií s odpovídajícím souřadnicovým systémem a pro rychlost také příslušné prostorové indexy.

Výsledkem je tabulka totožná s mateřskou tabulkou 6 a upravená tabulka 7 (a další tabulky s dlaždicemi), které jsou rozšířeny o sloupce *rid*, který je primárním klíčem a *rast*, který obsahuje geometrie v nativním formátu PostGIS raster.

Tento způsob uložení umožňuje manipulaci s rastry pomocí všech funkcí, kterými PostGIS raster disponuje. Na druhou stranu byla vytvořena duplicita dat v databázi, která ale přináší mnoho výhod zmiňovaných v kapitole 3.

5.2.4. Publikace mapové služby

Publikace mapové služby byla nejtěžším úkolem, protože není dokumentována žádná možnost, jak vypublikovat datový zdroj GeoServeru s připojením na PostGIS raster automaticky bez zásahu přímo do souborového systému, kde je nainstalovaný mapový server. Cílem bylo použít REST API GeoServeru ovládané skrze Javu. Protože dokumentace REST API (GeoServer, 2012 [b]) i další zdroje popisují pouze možnost vytvoření 3 následujících typů datových zdrojů

v GeoServeru (*geotiff*, *worldimage* a *imagemosaic*), ale nikoliv IMJDBC typu, bylo potřeba nalézt jinou cestu.

Po prostudování zdrojového kódu GeoServeru a testování byla nalezena možnost, kdy postačovalo rozšíření podporovaného typu *imagemosaic* na tvar *imagemosaicjdbc* a použít stejnou operaci REST API, která je využívána pro vytvoření podporovaných typů datových zdrojů. Příkaz CURL vytvářející datový zdroj IMJDBC (příloha 5).

Dále jsou používány operace REST API, které jsou již plně podporované.

Dílčím problémem, který zde nastal (ve verzi GeoServer 2.1.4), byl výskyt bílých ploch okolo publikovaných rastrů. Samotný rastr byl vypublikován správně, ale veškeré jeho okolí vyplňovala bílá barva, která překrývala ostatní vrstvy připojené v klientu. Řešením bylo přidání parametru průhlednosti tzv. *Output transparent color* do XML, které přijímá REST API GeoServeru při vytváření vrstvy. Tento parametr nebyl dostupný skrze GUI GeoServeru, dokud se nepoužila publikace vrstvy skrze REST API (příloha 6). V GUI se tento parametr objeví až po prvním založení vrstvy s tímto parametrem přes CURL.

5.2.5. Uživatelské kompozice

Uživatelské kompozice nejsou přímo v aplikačním kódu řešeny, protože jsou závislé na uživateli. Uživatelské účty a správa uživatelů v tomto konceptu ale nejsou řešeny. Dalším problémem je značně rozsáhlá problematika WMC, která by musela být řešena jak na straně klienta (vytvoření WMC dokumentu), tak na straně databáze (uložení WMC dokumentu).

Tento problém je v této práci vyřešen pomocí skupin vrstev, které umožňuje ukládat GeoServer pomocí REST API. Seznam vrstev lze získat pomocí dotazu na webovou stránku <http://localhost:8082/geoserver/rest/layers> a publikovat skupinu vrstev pomocí nástroje CURL z příkazové řádky jak ukazuje příloha 7.

Obě tyto operace vyžadují ověření uživatele.

5.2.6. Výstup

Výstupem aplikace jsou metadatové záznamy v GeoNetwork opensource validní podle ISO 19115. Zda budou data validní, záleží na metadatovém XML, které do aplikace vstupuje – GeoNetwork metadata nevaliduje. Operace validování může být zapnuta v případě používané operace GeoNetwork *metadata.insert*. Metadata obsahují informace zjištěné z rastrů a informace o možných dostupných elektronických zdrojích, které byly při běhu aplikace vytvořeny a publikovány do mapového serveru.

Dalším výstupem jsou mapové služby vypublikované GeoServerem a to ve více podobách, mapový server totiž nezakládá jednotlivé služby (WMS, WFS, KML, atd.) samostatně, ale všechny tyto služby jsou založené hromadně, pokud je vypublikována daná vrstva (služba).

Tento přístup je výhodný jak z pohledu administrátora, tak i z pohledu koncového uživatele. Vyžaduje méně konfiguračních prací, ale uživatel má více možností, v jakém formátu data získat.

Mimo data publikovaná mapovým serverem jsou dostupná rastrová data, která jsou uložena v interním rastrovém formátu PostGIS raster, která jsou připojitelná do různých GIS klientů v jejich surové podobě a je tak zpřístupněna možnost s daty pracovat na úrovni databáze a funkcí, které pro práci s rastry zprostředkovává.

5.3. Testování

Testování probíhalo nad různými typy dat (tabulka 8). V první řadě byly testovány staré mapy z mapové sbírky Přírodovědecké fakulty UK. Testy dopadly velmi dobře, a odezvy jsou vynikající, což reflektuje tabulka 9. Velmi podobně dopadly testy u výřezů z OSM (Open Street Maps) a přehledové mapy ČR 1: 500 000 (ČÚZK) i dat SRTM.

Tabulka 8: Testovaná data

Typ dat	Rozlišení [px]	Velikost [Mb]	Typ dat
Výřez OSM	1060x588	0,4	PNG
Výřez přehledové mapy 1: 500 000 (ČÚZK)	2445x924	1,9	PNG
Mapová sbírka 2	3509x1404	5,1	JPEG
Mapová sbírka 1	2369x1497	6,0	JPEG
SRTM data	6000x6000	68,7	GeoTiff
QuickBird data	4096x4096	128,0	GeoTiff

Zdroj: vlastní

Tabulka 9: Výsledky testování časových odezev v sekundách

Vrstva/Velikost požadavku [px]	50x50	150x150	500x500	1000x1000
Výřez OSM	0,123	0,147	0,239	0,447
Výřez přehledové mapy 1: 500 000 (ČÚZK)	0,139	0,196	0,337	0,596
Mapová sbírka 2	0,117	0,241	0,751	1,225
Mapová sbírka 1	0,119	0,243	0,883	1,072
SRTM data	0,130	0,188	0,492	1,414

Zdroj: vlastní

Testování bylo provedeno vždy sto dotazy GetMap na mapovou službu WMS o různém mapovém rozsahu (*bounding box*) tak, aby bylo vyloučeno „cachování“ a ve 4 různých velikostech (50×50, 150×150 a 500×500 a 1000×1000 pixelů). Test probíhal pouze z jednoho prostředí, proto mohou být výsledky zatíženy některými chybami na síti. Pro korektnější výsledky by bylo nutné provést mnohem více testů pomocí některé z testovacích aplikací, např. Apache JMeter na více prostředích a simulací zatížení větším množstvím uživatelů. Tyto výsledky byly testovány pomocí vlastní aplikace WMS Downloader, která stahuje definované čtverce ze zvolené WMS v oblasti definovaného mapového rozsahu.

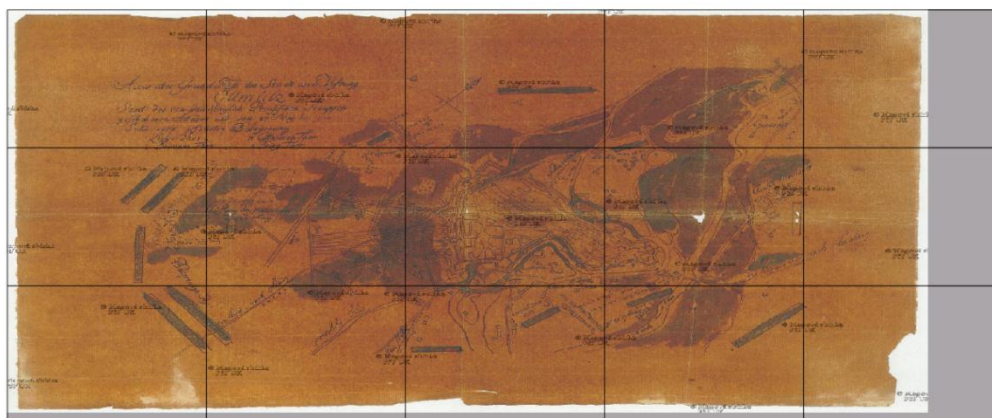
Problém nastal při testování dat z družice QuickBird, kdy tato data se podařilo naimportovat do databáze (kde jsou naimportována správně), podařilo se je i vypublikovat, ovšem výsledkem je pouze černý čtverec. Tento problém vzniká v souvislosti s tím, že funkce *ST_AsTIFF* nebo *ST_AsPNG*, které převádí data z nativního formátu rastru PostGIS do binární podoby, není schopna roztáhnout histogram tak, aby byl obraz pro službu správně interpretovatelný pro GeoServer.

V tomto místě byl odhalen při testování problém, kdy IMJDBC plugin je postavený nad Java Advanced Image (JAI) API, které podporuje pouze několik základních formátů, jako jsou BMP, FlashPix, GIF, JPEG, PNG, PNM, TIFF a WBMP (Oracle Java, c2010). Proto je třeba zajistit již na vstupu, aby data byla připravená pro publikaci mapovými službami (obsahovala pásma RGB nebo RGBA).

Zde nastává otázka, zda pokud rastry na vstupu takto modifikujeme, má cenu držet jejich nativní podobu v PostGIS. Data, která jsou takto modifikována, již nejsou vhodná pro analýzy, které vyžadují původní data. V tomto případě by bylo vhodnější rozdělit datové struktury na „publikační schéma“, které by obsahovalo pouze binární data pro publikaci služeb pomocí GeoServeru a na schéma „analytické“, které by obsahovalo původní data, přístupná pro analytické účely a zobrazení v klientech GIS přímo z databáze, včetně zpřístupnění všech možností, které poskytuje PostGIS raster.

Tento problém by bylo možné vyřešit situací, kdy by pro publikaci mapových služeb byly vytvářeny modifikované rastry v nějakém z podporovaných formátů JAI, ve výsledné podobě, která se má zobrazit ve vypublikované mapové službě. Takový přístup ovšem znehodnocuje samotná data, která jsou uložena v nativním rastrovém formátu PostGIS. Přijďme o možnosti PostGIS raster nad původními daty.

Během testování se projevil ještě jeden problém. Některé vypublikované vrstvy mají na své hraně černý rám. Tento problém je způsobem generováním dlaždic při importu rastrů pomocí nástroje *raster2pgsql*. Importovací nástroj vygeneruje dlaždice o definované velikosti, kdy šířka rastru nemusí být dělitelná beze zbytku velikostí dlaždice a poslední vygenerovaná dlaždice přesahuje rastr. Rastr je pak naimportován tak, že tam, kde poslední dlaždice přesahuje rastr je přidána tzv. *nodata* hodnota, která právě vytváří černou hranu ve vypublikovaném rastru. Tato situace je nastíněna na obrázku 14 v podobě šedivé barvy na spodním a pravém okraji.

Obrázek 12: Přesahy rastru vytvářející rám okolo publikovaného rastru

Zdroj: vlastní

Tato situace by byla možná vyřešit některým z následujících kroků.

1. Generovat dynamicky dlaždice podle velikosti rastru tak, aby vždy pokryly přesně celý rastr – toto nelze vždy splnit.
2. Přidat alfa kanál do importovaného rastru – toto ale neřeší příčinu problému.
3. Řešit situaci po importu dat přímo v databázi a využít funkcí PostGIS a tyto hodnoty odstranit z příslušných dlaždic přímo na úrovni databáze.
4. Vrátit se k původnímu importnímu nástroji IMJDBC, který používá k importu rastrů funkci *gdal_retile* v souborovém systému, která ukončí rastr v místě, kde rastr skutečně končí. Poslední řada dlaždic ve svislém i vodorovném směru je pak menší než ostatní dlaždice a tento nástroj modifikovat, tak aby vytvářel i nativní formát rastru v PostGIS.
5. Bylo objeveno, že směrem k vývojářům PostGIS je vznesen požadavek, aby *raster2pgsql* umožňoval nahrání dat tak, aby vytvářené dlaždice přesně kopírovali rastr a nepřesahovali mimo něj. Není specifikována verze, kde by toto mělo být hotové (PostGIS Ticket #826, 2012).

Protože tato chyba byla odhalena až při finálním testování, je zatím problém řešen postupem v bodě 2.

Data Meris nebyla testována, protože v dubnu 2012 bylo po 10 letech ztraceno spojení s družicí Envisat.

Kompletní zdrojový kód aplikace včetně komentářů a testovaných dat je dostupný na CD s elektronickou verzí práce.

KAPITOLA 6

Diskuze

Podařilo se vytvořit funkční aplikaci, která zajišťuje publikaci metadat a dat z databázového systému PostgreSQL za pomoci technologií mapového serveru GeoServer a metadatového katalogu GeoNetwork opensource.

Vzhledem k šíři této práce vyplynulo během zpracování několik otázek k diskuzi, které by měla tato kapitola pojmout a navrhnout jejich řešení.

Databázová platforma PostgreSQL s prostorovou nadstavbou PostGIS se ukázala jako velmi silný nástroj, se kterým je možné při velmi dobrých odezvách poskytovat připravené podklady mapovému serveru. Avšak na začátku je třeba specifikovat a rozdělit si jasně, k čemu data mají být použita a na základě tohoto rozhodnutí stanovit databázové struktury. MTDTRasPub nyní vypublikuje nativní formát rastru do PostGIS, tak zároveň binární podobu tohoto rastru pomocí funkcí PostGIS raster, kterou čte posléze GeoServer. Tento přístup byl zamýšlen tak, že obě struktury, jak binární data i nativní formát rastru jsou na jednom místě a v identických strukturách (dlaždice a pyramidy). Tento přístup se zdál velmi výhodným v tom, že je možné publikovat jak data pro GeoServer, tak využívat formátu PostGIS pro práci v GIS klientech a pro analytické práce, exporty a další funkce, které PostGIS společně s GDAL knihovnou poskytuje. Ovšem problematickým místem se ukázalo použití Java Advanced Image (JAI) API. Není možné připravit všechny formáty rastrů, které podporuje PostGIS jako nativní (příloha 2), do binární podoby (uložení BLOB), ze které čte data právě JAI API a poskytuje data GeoServeru (kapitola 5.3.). Problémem jsou rastry, které nemají obvyklá pásma RGB nebo RGBA a mají více pásem. Pak funkce PostGIS raster, které vytvářejí binární formát obrazu, neví jak správně interpretovat obraz pro mapovou službu. Proto by tento problém musel být řešen specificky pro každý typ importovaných rastrů, kdy se na úrovni databáze nechají ovlivnit některé parametry při převodu z nativního formátu do binárního. Výsledkem binární podoby rastru musí ovšem vždy být jeden z podporovaných formátů JAI API (kapitola 5.3.).

V důsledku tohoto problému se jeví jako lepší řešení rozdělit databázi na 2 schémata. Jedno z nich by bylo „publikační“, které by obsahovalo pouze binární data pro GeoServer, která by

vznikala z předpřipravených „náhledových“ rastrů, které by splňovaly požadavky formátů JAI API. Druhé schéma by bylo „analytické“, které by obsahovalo původní rastry v nativním formátu PostGIS a sloužilo by k analytickým úlohám, exportům a dalším operacím s původními daty, které umožňuje PostGIS a GDAL. Tato úprava by znamenala jen minimální úpravy na aplikaci.

Během vytváření práce byl na straně GeoServeru a hlavně knihovny GeoTools vidět velký vývoj, protože předešlý diskutovaný problém by šlo dnes vyřešit použitím nové knihovny GeoTools pro GeoServer. Ta umožňuje čtení rastrů přímo z nativního formátu PostGIS. Tím pádem odpadá nutnost převodu na binární tvar v databázi a s tím spojené duplikování dat a zároveň se s tímto zpřístupňují všechny možnosti PostGIS raster pro publikované vrstvy (GeoTools, c2012). V tomto místě by se také uplatnil přístup k importu rastrů, který byl v této práci navržen a realizován, kdy by postačovalo pouze zrušení vytváření binární podoby dat z rastru.

V případě černých hran okolo rastru je problémem standardní importní nástroj pro rastry *raster2pgsql*, který nepodporuje možnost načtení tak, aby krajní dlaždice nepřesahovaly přes rastry. Na toto téma byl nalezen otevřený „ticket“ (PostGIS Ticket #826, 2012), kde je v plánu rozšíření *raster2pgsql* o generování dlaždic tak, aby pokrývaly vždy přesně celý rastr. Poslední dlaždice ve sloupci i v řádce by byla menší než ostatní a dlaždice by přesně pokrývaly rastr. Toto by vyřešilo problém s černými hranami v rastrech bez alfa kanálu.

Předešlé dvě úpravy by vyřešily zmiňované neduhy, kterými aplikace v současné verzi trpí, pokud nejsou na vstupu rastrová data, která obsahují průhledný alfa kanál.

Předchozí problémy jsou do jisté míry generovány technologií, která k rastrům v PostGIS přistupuje v případě mapového serveru GeoServer. V blízké době bude uplatněna nová knihovna GeoTools pro přímý přístup k nativním rastrům PostGIS, která by měla vyřešit část zmiňovaných problémů. Pokud se povede uskutečnit zmiňovaný vývoj *raster2pgsql*, byly by vyřešeny všechny zmiňované nedostatky.

Zajímavým porovnáním by také mohlo být srovnání s konkurenčním UMN MapServerem. Tato porovnání bývají velmi časově i technicky náročná. OSGeo provádí jednou ročně tzv. WMS benchmarking (OSGeo, 2011), kde dochází k porovnání výkonu open source mapových serverů, v letošním roce by se tak mohl objevit i test, který by porovnával výkon v případě rastrů uložených v PostGIS, který v roce 2011 ještě nebyl součástí testu.

Metadatové řešení využívá možností GeoNetwork opensource XML services, které umožňují ovládání celého metadatového katalogu, konkrétně jsou využívány služby *user.login*, *metadata.insert* a *user.logout*. Služba *metadata.insert* má několik vlastností, kterými lze ovlivnit vkládané metadatové záznamy například jejich skupinu, zda se mají validovat a také například, jak se má služba chovat k metadatovým záznamům, jejichž identifikátor již byl nalezen v databázi (vytvoření nového nebo přepsání stávajícího). Služba *metadata.insert* velmi usnadňuje nahrávání metadat do databáze, protože nemusí být hlídáno, zda metadata se stejným identifikátorem již existují.

Lepším přístupem v případě metadat, by bylo využít OGC standard CSW, který by v případě nasazení MTDTRasPub nad jinou metadatovou technologií měl zajistit kompatibilitu, pokud daná technologie splňuje OGC CSW. Pro testovací účely MTDTRasPub není implementace CSW zásadní vzhledem k faktu, že není mnoho dalších technologií, které by bylo možné na poli opensource použít.

Poslední částí řešení je publikace uživatelských mapových kompozic. Zde se přímo nabízí implementace standardu WMC, ale tento standard a WMC dokumenty jsou obvykle navázané na uživatelská oprávnění, která tato práce vůbec neřeší. Další nutnou podmínkou pro nasazení WMC je podpora jak na straně klienta, tak na straně databáze pro ukládání WMC dokumentů společně s uživatelskými oprávněními a vyřešení komunikační linky mezi WMC stranou databáze a klienta. Proto je toto řešení vyřešeno přes skupiny vrstev, které podporuje GeoServer a pro účely MTDTRasPub jsou dostačující alternativou.

Do budoucna by bylo vhodné vyřešit transformaci rastrových podkladů mezi souřadnicovými systémy na straně mapového serveru, kterými se tato práce nezabývá. Tato funkce by byla přínosná pro zobrazování různých typů dat v různých souřadnicových systémech.

Částečně se také projevuje fakt, že je tento program první verzí aplikace, kdy trpí některými neduhy jako je volání externích programů jako například CURL, které by se nechaly v budoucím vývoji nahradit knihovnamí programovacího jazyka Java.

KAPITOLA 7

Závěr

Společně s rostoucím množstvím dat v GIS, které zajišťuje neustálý rozvoj, větší dostupnost pořizovacích technologií a v neposlední řadě také sdílení dat v síti internet, je třeba řešit i otázky s tím spojené. Jedním z největších potenciálů GIS technologií je právě internet. Tento fakt si uvědomuje i tato práce a snaží se rozšířit a ukázat možnosti automatizace procesů jako je publikace mapových služeb a metadatových záznamů z rastrových podkladů z databázové platformy PostgreSQL využívající mapový server GeoServer a metadatový katalog GeoNetwork opensource.

V textu práce je popsán návrh a struktura řešení automatické publikace mapových služeb z rastrových dat uložených v PostgreSQL a jejich metadat. Pro uložení rastrů je využita poměrně nová technologie PostGIS raster, kdy jsou rastry ukládány přímo v databázi.

Funkčnost navržených postupů byla ověřena implementací vlastní aplikace a otestováním na reálných datech, jako jsou skenované obrazy starých map Mapové sbírky Přírodovědecké fakulty UK, výřez OSM, či vzorek dat družice QuickBird. Vytvořené služby je možné nejen využívat, ale také nalézt vyhledáváním v metadatech a to jak prostorově, tak atributově.

Je třeba kriticky poukázat na některé nedostatky, kterými jsou hlavně nemožnost publikace veškerých rastrových typů, kde jsou překážkou omezené možnosti technologií, které byly využity. V tomto případě se zde v plné míře projevuje výhoda open source technologií, kde některá omezení jsou již v řešení na straně vývojářů. Dále nedodržení některých OGC standardů, které byly obtížně začlenitelné do aplikace nebo vyžadovaly velmi náročné a zdlouhavé procesy pro jejich implementaci a nebyly pro vytvoření funkční verze zásadní.

Aplikace MTDTasPub bude dále vyvíjena společně s rozvojem technologií, které tato aplikace využívá, aby splnila očekávání, které na ni budou kladeny v případě, že se povede nasazení například pro účely zveřejňování historických děl mapové sbírky Přírodovědecké fakulty UK. Cílem rozvoje aplikace je také vyřešení diskutovaných bodů.

Seznam zdrojů informací

- ARÉVALO, J. *PostGIS WKT Raster. An Open Source alternative to Oracle GeoRaster*. konference FOSS4G Barcelona [online]., 2010 [cit. 2011-10-30]. Dostupné z URL: <<http://2010.foss4g.org/presentations/3814.pdf>>.
- AUJEZDSKÝ, J. *Právní aspekty volně šiřitelných počítačových programů: Open Source software*. [2009] In: *Root.cz* [online]. [cit. 2012-08-18]. Dostupné z URL: <<http://www.root.cz/specialy/licence/open-source-software/>>
- BARTUNOV, O.; SIGAEV, T. *SP-GiST – a new indexing framework for PostgreSQL*. In: PG Conf 2011 [online]. 2011 [cit. 2012-08-24]. Dostupné z URL: <http://www.pgcon.org/2011/schedule/attachments/197_pgcon-2011.pdf>
- BLASBY, D. *Building a spatial database in PostgreSQL*. In: Open Source Database Summit, 2001 [online]. 2001 [cit. 2012-08-28]. Dostupné z URL: <http://postgis.refractions.net/files/OSDB2_PostGIS_Presentation.ppt>
- CHEN, D; SHAMS, S.; CARMONA-MORENO, C.; LEONE, A. *Assessment of open source GIS software for water resources management in developing countries*, Journal of Hydro-environment Research, Volume 4, Issue 3, October 2010, Pages 253-264, ISSN 1570-6443, 10.1016/j.jher.2010.04.017. Dostupné z URL: <<http://www.sciencedirect.com/science/article/pii/S1570644310000511>>
- CURL. *Curl*. [online]. 2012 [cit. 2012-08-24]. Dostupné z URL: <<http://curl.haxx.se/>>
- DOELEMEN, H.; GARCÍA, J. *Mastering Advanced GeoNetwork*. In: konference FOSS4G Barcelona 2010 [online]., 2010 [cit. 2011-10-30]. Dostupné z URL: <http://geonetwork-opensource.org/_static/foss4g2010/FOSS4G_Mastering_Advanced_GeoNetwork.pdf>.
- EnterpriseDB. *Product Lifecycle PostgreSQL from EnterpriseDB* [online]. c2012 [cit. 2012-07-15]. Dostupné z URL: <<http://www.enterprisedb.com/products-services-training/products-overview/PostgreSQL-overview/supported-platforms-and-release-lif>>

- FLOWER, C. *MapServer and GeoServer (and tilecache) comparison serving Ordnance Survey raster maps*. [online]. 2012 [cit. 2012-08-21]. Dostupné z URL:
<<http://www.esdmdata.co.uk/mapserver-and-geoserver-and-tilecache-comparison-serving-ordnance-survey-raster-maps>>
- GRILL, S.; JEDLIČKA, J.; SCHNEIDER, M.; ŠTEFANOVÁ, E. *Archive and catalogue system for receiving satellite data as a part of academic SDI* [online]. Praha : IOS Press Books Online, 2010. 10 s. PřF UK Praha, Katedra aplikované geoinformatiky a kartografie. Dostupné z URL: <<http://www.booksonline.iospress.nl/Content/View.aspx?piid=16427>>. ISBN 978-1-60750-493-1.
- GDAL. *Software Using GDAL*. [online]. 2012 [cit. 2012-08-20]. Dostupné z URL:
<<http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal>>
- GeoNetwork opensource. *Retrieve metadata services* [online]. c2010 [cit. 2011-11-16]. Dostupné z URL: <http://geonetwork-opensource.org/latest/developers/xml_services/metadata_xml_services.html>.
- GeoServer. *GeoServer User Manual: Storing a coverage in a JDBC database*. [online]. 2012 [a] [cit. 2012-08-24]. Dostupné z URL:
<<http://docs.geoserver.org/latest/en/user/index.html>>
- GeoServer. *GeoServer User Manual: REST Configuration API Reference*. [online]. 2012 [b] [cit. 2012-08-24]. Dostupné z URL: <<http://docs.geoserver.org/latest/en/user/restconfig/rest-config-api.html>>
- GeoTools. *Postgis Raster Plug in*. [online]. c2012 [cit. 2012-08-24]. Dostupné z URL:
<<http://docs.geotools.org/latest/userguide/library/coverage/pgraster.html>>
- GNU Operating System. *Filosofie projektu GNU. Definice svobodného software* [online]. c1996-1999 [cit. 2012-08-19]. Dostupné z URL: <<http://www.gnu.org/philosophy/free-sw.html>>
- JEŽEK, J. *Webová aplikace pro operace se souřadnicovými systémy*. [online]. 2006 [cit. 2012-08-20]. Dostupné z URL:
<http://www.vugtk.cz/odis/sborniky/jine/geos06/paper/32_jezek_j/paper/32_jezek_j.pdf>
- KAFKA, Š. *Metadatový portál a katalogové služby*. In: GIS Ostrava 2008 [online]. 2008 [cit. 2012-08-26]. Dostupné z URL:
<http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2008/sbornik/Lists/Papers/028.pdf>
- MANEIRO, A. ... [et al.]. *Análisis de adopción, actividad y participación en aplicaciones SIG Libres: un estudio sobre GRASS, Quantum GIS y gvSIG basado en indicadores*. In: Jornadas de SIG Libre [online]. 2011 [cit. 2012-08-20]. Dostupné z URL:
<<http://www.sigte.udg.edu/jornadassiglibre2011/uploads/articulos/art37.pdf>>

- MUELLER, Ch. *Image Mosaicing Pyramidal JDBC Plugin*. [online]. 2009 [cit. 2012-08-26]. Dostupné z URL:
<<http://docs.codehaus.org/display/GEOTDOC/Image+Mosaicing+Pyramidal+JDBC+Plugin>>
- NOVOTNÁ, E. *Technologie pro zpřístupnění mapových sbírek ČR : šance pro oživení Mapové sbírky PŘF UK*. I-FORUM [online]. Praha : UK, 2011 [cit. 2012-08-02]. Dostupné z URL:
<<http://iforum.cuni.cz/IFORUM-10507.html>>. ISSN 1214-5726.
- OBE, R.; HSU, L. *PostGIS 2.0 3D and Raster support enhancements*. [online]. 2011 [cit. 2012-08-24]. Dostupné z URL:
<http://www.postgis.us/downloads/ncgis2011/NCGISSDBPostGIS20_2011.pdf>
- OBE, R.; HSU, L. *PostGIS in Action*. Stamford: Manning Publications Co., 2011 [b], 425 s. ISBN 9781935182269.
- OGC. *About OGC*. [online]. c1994-2012 [cit. 2012-08-27]. Dostupné z URL:
<<http://www.opengeospatial.org/ogc>>
- Open Source GIS Software: *FOSS4G Software List*. [online]. 2012 [cit. 2012-08-19]. Dostupné z URL:
<http://sourceforge.net/userapps/mediawiki/mentaer/index.php?title=FOSS4G_Software_List>
- Open Source Initiative. *Open Source Licenses by Category*. [online]. 2012 [cit. 2012-08-18]. Dostupné z URL: <<http://opensource.org/licenses/category>>
- OSGeo. *WMS Benchmarking 2011*. OSGEO. [online]. 2011 [cit. 2012-08-21]. Dostupné z URL:
<<http://svn.osgeo.org/osgeo/foss4g/benchmarking/wms/2011/presentation/>>
- OSGeo. *Open Source Geospatial Foundation*. OSGEO. [online]. 2012 [cit. 2012-08-18]. Dostupné z URL: <<http://www.osgeo.org/>>
- OSGeo4W. *OsGeo for Windows*. OSGeo4W [online]. 2012 [cit. 2012-08-25]. Dostupné z URL:
<<http://trac.osgeo.org/osgeo4w/>>
- OSS Aliance. *O open-source*. Společnost pro výzkum a podporu open source Fakulta jaderná a fyzikálně inženýrská. [online]. 2012 [cit. 2012-08-19]. Dostupné z URL:
<<http://www.oss.cz/cs/open-source>>
- Oracle. *Oracle Spatial* [online]. 2012 [cit. 2012-08-19]. Dostupné z URL:
<<http://www.oracle.com/cz/products/database/options/spatial/index.html>>
- Oracle Java. *Java Advanced Imaging API Home Page* [online]. c2010 [cit. 2012-08-19]. Dostupné z URL: <<http://java.sun.com/products/java-media/jai/iio.html>>
- OŽANA, R.; *Posouzení vlastností GeoNetwork Opensource a jeho uplatnitelnosti pro účely národního metaPortálu* [online]. 2007 [cit. 2011-11-16]. Dostupné z URL: <
<http://cs.scribd.com/doc/2165656/Posouzeni-vlastnosti-GeoNetwork-opesnource-a-jeho-uplatnitelnosti-pro-uely-narodniho-metaPortalu>>

- PGFoundry. *PgFoundry: PostgreSQL/EnterpriseDB Stack Builder: Project Info* [online]. 2007 [cit. 2012-07-16]. Dostupné z URL: <<http://pgfoundry.org/projects/stackbuilder/>>
- PostGIS. *PostGIS 2.0.2SVN Manual*. [online]. 2012 [cit. 2012-08-25]. Dostupné z URL: <<http://postgis.refrations.net/documentation/manual-2.0/>>
- PostGIS Ticket #826. [online]. 2012 [cit. 2012-08-19]. Dostupné z URL: <<http://trac.osgeo.org/postgis/ticket/826>>
- PostgreSQL. *PostgreSQL 8.0.26 Documentation*. [online]. 2005-01-19. 2005 [cit. 2012-07-15]. Dostupné z URL: <<http://www.PostgreSQL.org/files/documentation/pdf/8.0/PostgreSQL-8.0-A4.pdf>>
- RACINE, P. *Store, manipulate and analyze raster data within the PostgreSQL/PostGIS spatial database*. In: Sborník konference FOSS4G Denver [online]., 2011 [cit. 2011-10-30]. Dostupné z URL: <http://trac.osgeo.org/postgis/raw-attachment/wiki/WKTRaster/PierreRacine_FOSS4G-2011.pdf>.
- RAMSEY, P. *The State of Open Source GIS*. [online]. 2007 [cit. 2012-08-20]. Dostupné z URL: <<http://www.refrations.net/expertise/whitepapers/opensourcesurvey/survey-open-source-2007-12.pdf>>
- SKLENIČKA, R. *Interoperabilita v GIS podle specifikací OGC*. In Geoinformatics CTU 2006 [online]. Praha : České vysoké učení technické v Praze. Fakulta stavební. Katedra mapování a kartografie, 2006 [cit. 2011-10-30]. Dostupné z URL: <<http://maps.fsv.cvut.cz/~sklena/sklena/interoperabilitaOGC.pdf>>.
- Spatial database. In: Wikipedia: the free encyclopedia [online]. 2012. vyd. San Francisco (CA): Wikimedia Foundation, 2001-, 2012 [cit. 2012-08-19]. Dostupné z URL: <http://en.wikipedia.org/wiki/Spatial_database>
- STEINIGER, S.; BOCHER E. *An overview on current free and open source desktop GIS developments*. International Journal of Geographical Information Science [online]. 2009, roč. 23, č. 10, s. 1345-1370 [cit. 2012-08-19]. ISSN 1365-8816. DOI: 10.1080/13658810802634956. Dostupné z URL: <<http://www.tandfonline.com/doi/abs/10.1080/13658810802634956>>
- STEINIGER, S.; HUNTER, A.J.S. *Free and open source GIS software for building a spatial data infrastructure*. In E. Bocher and M. Neteler (eds): Geospatial Free and Open Source Software in the 21st Century: Proceedings of the first Open Source Geospatial Research Symposium, 2009, LNG&C, Springer, Heidelberg, pp. 247-261.
- STEINIGER, S; WEIBEL, R. *GIS software: a description in 1000 words*. In: Warf, B. Encyclopedia of geography. London, GB, [online]. 2010 [cit. 2012-08-20]. ISBN 978-1412-95697-0. Dostupné z URL: <http://www.zora.uzh.ch/41354/1/Steiniger_Weibel_GIS_Software_2010.pdf>
- SQLite. [online]. 2012 [cit. 2012-08-19]. Dostupné z URL: <<http://www.sqlite.org/about.html>>

- THON, L. *Lehký úvod do MongoDB*. [online]. 2010 [cit. 2012-08-20]. Dostupné z URL: <<http://www.abclinuxu.cz/clanky/programovani/lehky-uvod-do-mongodb>>
- TOPPER, M. *Comparison of the Oracle Locator and Oracle Spatial Feature*. [online]. 2006 [cit. 2012-08-20]. Dostupné z URL: <<http://www.mattpopper.com/2006/05/comparison-of-the-oracle-spatial-and-locator-features/>>
- VOHNOUT, P. *Portál pro staré mapy*. Plzeň, 2009. Diplomová práce. ZČU Plzeň. Vedoucí práce Doc. Ing. Václav Čada, CSc.

Seznam příloh

- Příloha 1 CD s elektronickou verzí práce, zdrojovým kódem aplikace a testovacími daty
- Příloha 2 Přehled rastrových formátů pro import do PostGIS raster
- Příloha 3 Ukázka konfiguračního souboru AppConfig.xml
- Příloha 4 Ukázka konfigurace mapové vrstvy
- Příloha 5 Část zdrojového kódu realizující založení GeoServer DataStore pro IMJDBC
- Příloha 6 Část zdrojového kódu realizující založení rastrové vrstvy GeoServer
- Příloha 7 Ukázka příkazu CURL pro založení skupiny vrstev GeoServer

Příloha 2: Přehled rastrových formátů pro import do PostGIS raster

Virtual Raster	Golden Software Binary Grid (.grd)
GeoTIFF	Golden Software 7 Binary Grid (.grd)
National Imagery Transmission Format	COSAR Annotated Binary Matrix (TerraSAR-X)
Raster Product Format TOC format	TerraSAR-X Product
ECRG TOC format	DRDC COASP SAR Processor Raster
Erdas Imagine Images (.img)	R Object Data Store
CEOS SAR Image	Portable Pixmap Format (netpbm)
CEOS Image	USGS DOQ (Old Style)
JAXA PALSAR Product Reader (Level 1.1/1.5)	USGS DOQ (New Style)
Ground-based SAR Applications Testbed File Format (.gff)	ENVI .hdr Labelled
ELAS	ESRI .hdr Labelled
Arc/Info Binary Grid	Generic Binary (.hdr Labelled)
Arc/Info ASCII Grid	PCI .aux Labelled
GRASS ASCII Grid	Vexcel MFF Raster
SDTS Raster	Vexcel MFF2 (HKV) Raster
DTED Elevation Raster	Fuji BAS Scanner Image
Portable Network Graphics	GSC Geogrid
JPEG JFIF	EOSAT FAST Format
In Memory Raster	VTP .bt (Binary Terrain) 1.3 Format
Japanese DEM (.mem)	Erdas .LAN/.GIS
Graphics Interchange Format (.gif)	Convair PolGASP
Graphics Interchange Format (.gif)	Image Data and Analysis
Envisat Image Format	NLAPS Data Format
Maptech BSB Nautical Charts	Erdas Imagine Raw
X11 PixMap Format	DIPEX
MS Windows Device Independent Bitmap	FARSITE v.4 Landscape File (.lcp)
SPOT DIMAP	NOAA Vertical Datum .GTX
AirSAR Polarimetric Image	NADCON .los/.las Datum Grid Shift
RadarSat 2 XML Product	NTv2 Datum Grid Shift
PCIDSK Database File	ACE2
PCRaster Raster File	Snow Data Assimilation System
ILWIS Raster Map	Swedish Grid RIK (.rik)
SGI Image File Format 1.0	USGS Optional ASCII DEM (and CDED)
SRTMHGT File Format	GeoSoft Grid Exchange Format
Leveller heightfield	Northwood Numeric Grid Format .grd/.tab
Terragen heightfield	Northwood Classified Grid Format .grc/.tab
USGS Astrogeology ISIS cube (Version 3)	ARC Digitized Raster Graphics
USGS Astrogeology ISIS cube (Version 2)	Standard Raster Product (ASRP/USRP)
NASA Planetary Data System	Magellan topo (.blx)
EarthWatch .TIL	SAGA GIS Binary Grid (.sdat)
ERMapper .ers Labelled	Kml Super Overlay
NOAA Polar Orbiter Level 1b Data Set	ASCII Gridded XYZ
FIT Image	HF2/HFZ heightfield raster
GRIdded Binary (.grb)	OziExplorer Image File
Raster Matrix Format	USGS LULC Composite Theme Grid
EUMETSAT Archive native (.nat)	Arc/Info Export E00 GRID
Idrisi Raster A.1	ZMap Plus Grid
Intergraph Raster	NOAA NGS Geoid Height Grids
Golden Software ASCII Grid (.grd)	

Příloha 3: Ukázka konfiguračního souboru AppConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <!--Absolutní cesta k adresari, ve kterém se nachází složka s rastry -->
  <absolutePath>c:\\users\\jakub\\Desktop\\</absolutePath>
  <!--Adresář s rastry -->
  <rasterDir>raster2import</rasterDir>
  <!--Cesta k datovému adresari GeoServeru -->
  <geoserverDataDir>C:\\Progr~\\GeoServer2.2.0\\data_dir\\data\\</geoserverDataDir>
  <!--GeoServer uživatelské jméno -->
  <geoserverUserName>admin</geoserverUserName>
  <!--GeoServer heslo -->
  <geoserverPassword>geoserver</geoserverPassword>
  <!--GeoNetwork uživatelské jméno -->
  <geonetworkUserName>admin</geonetworkUserName>
  <!--GeoNetwork heslo -->
  <geonetworkPassword>admin</geonetworkPassword>
  <!--Připojení k databázi pro publikaci rastru -->
  <postgresConection>jdbc:PostgreSQL://localhost/test</postgresConection>
  <!--PostgreSQL uživatelské jméno -->
  <postgresUserName>postgres</postgresUserName>
  <!-- PostgreSQL heslo -->
  <postgresPassword>password</postgresPassword>
  <!--URL GeoServeru -->
  <geoserverLocation>http://localhost:8082/geoserver</geoserverLocation>
  <!--URL GeoNetworku -->
  <geonetworkLocation>http://localhost:8080/geonetwork</geonetworkLocation>
</root>
```

Příloha 4: Ukázka konfigurace mapové vrstvy

```
<?xml version="1.0" encoding="UTF-8"?>
<mapConfig>
  <!--Titulek vrstvy - zatím není používán, použijte se jméno rastru -->
  <Title>Přehledová mapa 500000</Title>
  <!--Workspace, kam se má rastr vypublikovat -->
  <Workspace>Testovací</Workspace>
  <!--EPSG kód vrstvy -->
  <EPSG>4326</EPSG>
</mapConfig>
```

Příloha 5: Část zdrojového kódu realizující založení GeoServer DataStore pro IMJDBC

```
curl -u "+geoserverUserName+": "+geoserverPassword+" -v -XPUT -H "Content-type:
text/xml\" -d @" + TmpMapConfigFile + " "+geoserverLocation+"/rest/workspaces/" +
Workspace + "/coveragestores/" + CoverageStore + "/file.imagemosaicjdbc"
```

Příloha 6: Část zdrojového kódu realizující založení rastrové vrstvy GeoServer

```
curl -u "+geoserverUserName+": "+geoserverPassword+" -v -XPUT -H "Content-type:
text/xml\" "
+ "-d \"<coverage><enabled>true</enabled><name>" + LayerName + "</name><title>" +
LayerName +
"</title><parameters><entry><string>OutputTransparentColor</string><string>#FFFFFF</stri
ng></entry><entry><string>BackgroundColor</string><string>#FFFFFF</string></entry></para
meters></coverage>\" "
+ geoserverLocation+"/rest/workspaces/" + Workspace + "/coveragestores/" + CoverageStore
+ "/coverages/"+LayerName+".xml"
```

Příloha 7: Ukázka příkazu CURL pro založení skupiny vrstev GeoServer

```
curl -u admin:geoserveradmin -v -XPOST -H "Content-type: text/xml" -d
"<layerGroup><name>TestGroup</name><layers><layer>roads</layer><layer>parks</layer></lay
ers></layerGroup>" http://localhost:8082/geoserver/rest/layergroups.
```